IEC 61850: a safety and security analysis in industrial multiprotocol networks

Luca Rocca

Supervisor: **Prof. Paolo Pinceti**

Department of Naval, Electrical, Electronics and Telecommunication Engineering

University of Genoa

This dissertation is submitted for the degree of

Doctor of Philosophy in Electrical Engineering – XXX cycle

University of Genoa May 2019

CONTENTS

1. INTRODUCTION

Nowadays, Ethernet is the most popular technology in digital communication thanks to its flexibility and worldwide spread. This is the reason why the main industrial communication protocols today are based on Ethernet. Everybody says that Ethernet supports a large amount of different protocols, but only accurate laboratory tests can make this assumption true. Tests are performed on a hybrid network using three protocols: Profinet, IEC 61850 and TCP/IP. The combination of these three protocols represents an ideal industrial application where process automation, substation automation and general purpose data sharing interact. However, a shared network can be the cause of a drop in terms of safety and security in the industrial plant network.

Safety has always played an important role in the life of human beings and the environment. The safety of process control systems is standardized in IEC 61508 and IEC 61784-3, but this is not the same in the area of substation automation system. Several tests are performed to prove if IEC 61850 (the standard protocol for substation automation) meets the requirements stated in IEC 61508 and if it can be used for safety-related functions[1][2][12].

Security issues for industrial plants have become increasingly relevant during the past decade as the industry relied more and more on communication protocols. This work examines the security issues for IEC 61850 addressed by IEC 62351-6 providing a deepening for a secure GOOSE communication. The major issue implementing such a standard remains the computational power requested by the SHA algorithm in low-powered devices. As no manufacturer has made available a device implementing secure GOOSE communication yet, this solution is discussed only from a theoretical point of view. After that, it is presented a security test on the GOOSE communication during which the security issues of such a communication are exploited. This test aims to show what consequences may occur when a packet artfully created is injected within the IEC 61850 network. In the last part of this section, some countermeasures to mitigate such an issue are provided.

2. HYBRID NETWORKS PERFORMANCE MEASURES

A hybrid network is an industrial multiprotocol network that shares a common physical infrastructure. It means that all the protocols needed in a typical industrial application runs on the same bus. The tested hybrid networks are based on an Ethernet infrastructure, and the industrial protocols chosen are three: IEC 61850 (electrical substation standard), Profinet IO (process automation standard) and TCP/IP (typical for general purpose devices). It must be underlined that all the protocols are native, that no gateway or interface is required for a full interconnection of each sub-system.

A hybrid network offers several benefits to users, such as:

- very simple network architecture;
- shared standard cable infrastructure;
- use of standard network components;
- integration of other application (streaming video for security cameras, ERP, etc.);
- each protocol runs in native mode, without the need of gateways or complex and time-consuming interfaces.

The tested architectures are very significant, since all industrial applications need the coordinated interaction between the process control, the electrical distribution supervision, and the security systems. Automation is made of control, supervision, and configuration of field intelligent devices. Each function requires hardware where software runs, and it can be either distributed or centralized. The combination of all these components and locations gives origin to a bunch of solutions, each one with pros and cons (and some positively impractical).

In order to test the integration between the used industrial protocols, a SCADA that can monitor and control both the Profinet and the IEC 61850 networks. The selected SCADA is Zenon provided by COPADATA, one of the first software companies that has integrated IEC 61850 in its software.

The implemented Ethernet-based network contains all the components that typically constitute an automation system:

- SCADA;
- PLC and/or Soft PLC;
- Ethernet Switch;
- Digital I/O devices interfaced via Profinet IO;
- Intelligent Electrical Devices interfaced via IEC 61850.

The tested automation architectures are mainly two. In the first the protocols share the physical infrastructure but there isn't interaction between different devices: the SCADA receives the variables from the Profinet and the IEC 61850 networks but the control of the two protocols is separated. On the other hand, the second configuration permits to create logics with variables from both the communication protocols, thanks to Straton, the Soft-PLC integrated in Zenon.

The procedures, the setup and the results of the tests will be explained focusing on two main aspects: the time performance of the integrated protocols and the security of the network under test, following the guidelines of PI consortium.

2.1 HYBRID NETWORKS

2.1.1 OVERVIEW

An industrial Ethernet-based hybrid network is a network for industrial automation on which more than one protocol runs on the same physical infrastructure. Today hybrid networks are the best solution to implement a network in an industrial plant. The main advantage is the use of a unique physical infrastructure, that means simplified and cost-effective installation, unified maintenance, training, spare parts and flexibility in term of system extensions.

Sharing the same physical infrastructure is only possible using Ethernet that guarantee the necessary bandwidth the protocols need to not interfere on the real time performances of the other protocols.

2.1.2 INTEGRATION OF PROCESS AUTOMATION, POWER AUTOMATION AND GENERAL PURPOSE APPLICATIONS

Traditionally, any industrial plant consists in several areas each containing a part of the process (see Figure 1):

- Process control: includes instrumentation, safety systems and controllers but also “process electrification” devices such as low voltage drives and motors. Here these devices typically communicate using a variety of fieldbus technologies including Profibus, Foundation Fieldbus, Hart, Profinet and ModBus.
- Power automation: corresponds to Substation Automation. It consists in MV and HV power equipment including protective relays, power meters, drives and motors. These devices are connected using ad hoc fieldbuses such as, ModBus/TCP, IEC 60870, DNP3/TCP and IEC 61850.
- General purpose applications: such as the security system or the management system of the plant. The typical network used to fulfil these “office” functions is Ethernet TCP/IP.

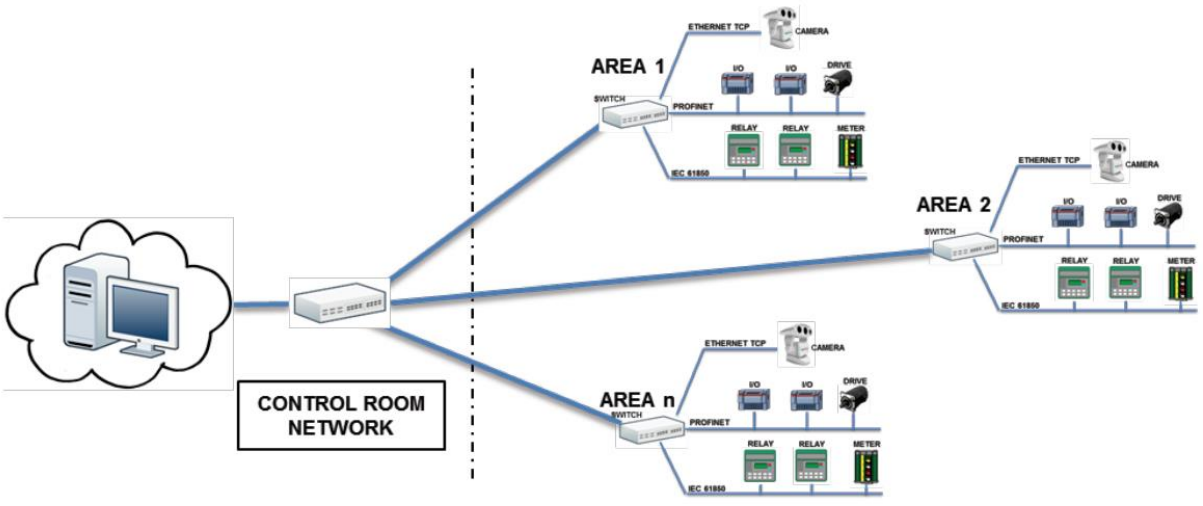


Figure 1 Typical structure of an industrial plant communication system

The systems that serve process automation and electrical automation are separated or interfaced with hard-wired connections (Figure 2). The mean of a hybrid Ethernet-

based network is to connect these two aspects of the automation of the entire plant, creating a single automation environment.

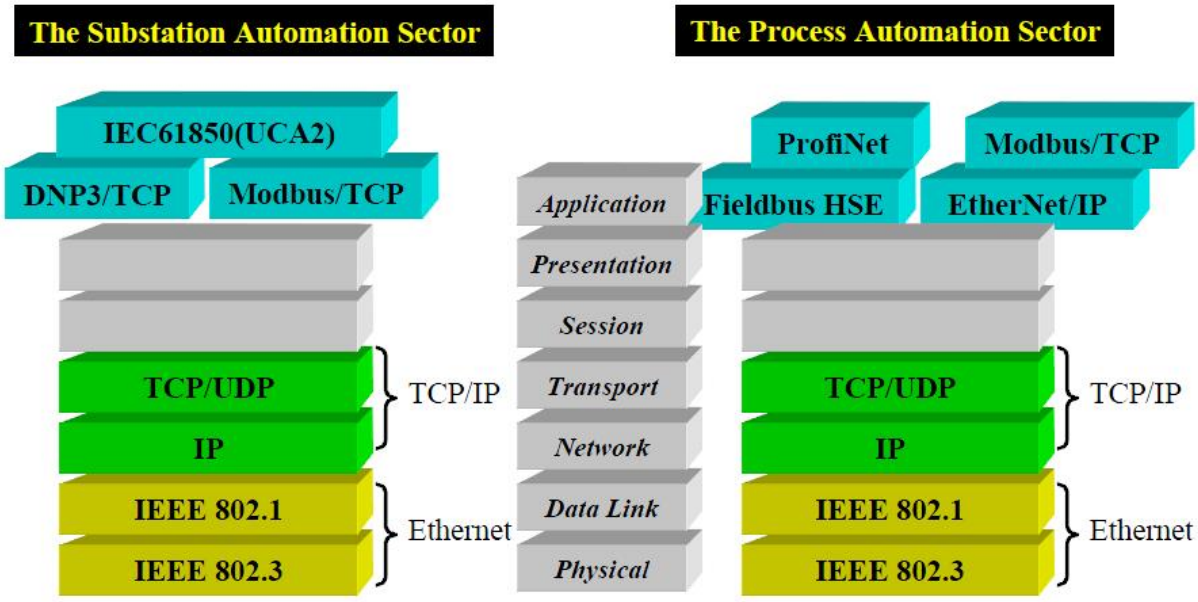


Figure 2 Substation Automation vs Process Automation

2.2 ETHERNET OVERVIEW

In the last twenty years Ethernet has become the main technology in personal, office and industrial applications thanks to the world trend that aim to web applications and integration. Nowadays Ethernet can be used both for control room network and for communication between field devices; it allows to reduce costs and to have a unique physical way that goes from the top to the bottom of the automation pyramid: from the strategic data exchange level to the process level. Furthermore, Ethernet enables a simple integration with Internet that can lead to an easy use of decentralized control systems. Integrating information technology into automation opens to significantly better communication options between automation systems, extensive configuration and diagnostic possibilities, and network-wide service functionality.

At the beginning there were two problems using the Ethernet technology for industrial protocols: the specifications of Ethernet were not complaint for industrial applications

on field (not deterministic), and other fieldbus were too much ingrained in the market. To fill the lack of a deterministic, or real time, communication on Ethernet, new Ethernet-based protocols have been developed and today the industrial world is moving from traditional fieldbus to Ethernet-based fieldbus.

Referring to the ISO/OSI model, the Ethernet protocol collocate itself at the layer called Data Link. Such a layer, according to the IEEE 802, can be further split into two levels: an upper sublevel called LLC (Logical Link Control) and a lower sublevel called MAC (Media Access Control) (Figure 3).

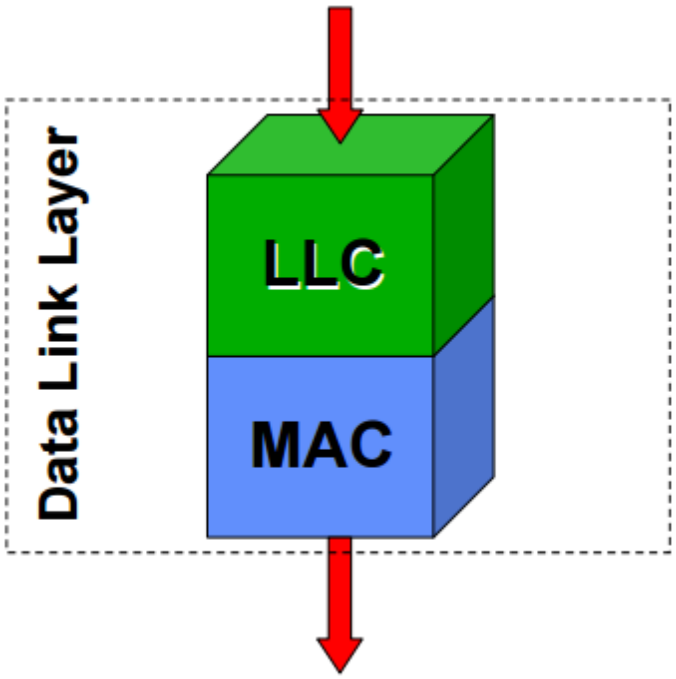


Figure 3 Data Link Layer split in MAC and LLC

While LLC is responsible for handling multiple “Layer3” protocols (multiplexing/de-multiplexing) and link services like reliability and flow control, the MAC is responsible for framing and media access control for broadcast media.

Below is a list of the primary responsibility of LLC sublayer:

- Network Layer protocol Multiplexing/De-Multiplexing: Interfacing with the Network (Layer3) above by doing L3 protocol multiplexing/de-multiplexing. On receiving a frame from the physical layer below, the LLC is responsible for

looking at the L3 Protocol type and handing over the datagram to the correct L3 protocol (de-multiplexing) at the network layer above. On the sending side, LLC takes packets from different L3 protocols like IP, IPX, ARP etc., and hands it over to the MAC layer after filling the L3 protocol type in the LLC header portion of the frame (multiplexing).

- Logical Link Services: LLC can optionally provide reliable frame transmission by the sending node numbering each transmitted frame (sequence number), by the receiving node acknowledging each received frame (acknowledgment number) and by the sending node retransmitting lost frames. It can also optionally provide flow control by allowing the receivers to control the sender's rate through control frames.

Below is a list of the primary responsibility of MAC sublayer:

- Framing/De-Framing and interaction with the physical layer: On the sending side, the MAC sub-layer is responsible for the creation of frames from network layer packets by adding the frame header and the frame trailer. While the frame header consists of layer2 addresses (known as MAC address) and a few other fields for control purposes, the frame trailer consists of the CRC/checksum of the whole frame. After creating a frame, the MAC layer is responsible for interacting with the physical layer processor (PHY) to transmit the frame. On the receiving side, the MAC sub-layer receives frames from the PHY and is responsible of accepting each frame, by examining the frame header. It is also responsible for verifying the checksum to conclude whether the frame has come uncorrupted through the link without bit errors. Since checksum computation and verification are compute intensive tasks, the framing/de-framing functionality is done by dedicated piece of hardware;
- Collision Resolution: On shared or broadcast links, where multiple end nodes are connected to the same link, there must be a collision resolution protocol running on each node, so that the link is used cooperatively. The MAC sub-layer is responsible for this task and it is the MAC sub-block that implements standard collision resolution protocols like CSMA/CD, CSMA etc. For half-duplex links, it

is the MAC sub-layer that makes sure that a node sends data on the link only during its turn.

Carrier Sense Multiple Access (CSMA) is a MAC protocol in which a node verifies the absence of other traffic before transmitting on a shared transmission medium, e.g. an electrical bus, or a band of the electromagnetic spectrum. A transmitter attempts to determine whether another transmission is in progress before initiating a transmission using a carrier sense mechanism. That is, it tries to detect the presence of a carrier signal from another node before attempting to transmit. If a carrier is sensed, the node waits for the transmission in progress to end before initiating its own transmission. Using CSMA, multiple nodes may send and receive on the same medium. All other nodes connected to the medium generally receive transmissions by one node.

Four different types of CSMA are defined:

- 1-persistent: It is an aggressive transmission algorithm. When the transmitting node is ready to transmit, it senses the transmission medium for idle or busy. If the medium is idle, then the node transmits immediately. If the medium is busy, then the node senses the transmission medium continuously until it becomes idle, then transmits the message (a frame) unconditionally (i.e. with probability = 1). In case of a collision, the sender waits for a random period and attempts the same procedure again. This method is used in CSMA/CD systems including in Ethernet.
- Non-Persistent: it is a non-aggressive transmission algorithm. When the transmitting node is ready to transmit data, it senses the transmission medium for idle or busy. If it is idle, then the node transmits immediately. If the medium is busy, then the node waits for a random period (during which it does not sense the transmission medium) before repeating the whole logic cycle (which started with sensing the transmission medium for idle or busy) again. This approach reduces collision, results in overall higher medium throughput but with a penalty of longer initial delay compared to 1–persistent.
- P-Persistent: This is an approach between 1-persistent and non-persistent CSMA access modes. When the transmitting node is ready to transmit data, it

senses the transmission medium for idle or busy. If idle, then it transmits a frame with probability p. If busy, then it senses the transmission medium continuously until it becomes idle, then transmits with probability p. If the node does not transmit (the probability of this event is 1-p), it waits until the next available time slot. If the transmission medium is still not busy, it transmits again with the same probability p. This probabilistic hold-off repeats until the frame is finally transmitted or when the medium is found to become busy again (i.e. some other node has already started transmitting). In the latter case the node repeats the whole logic cycle (which started with sensing the transmission medium for idle or busy) again.

- O-Persistent: Each node is assigned a transmission order by a supervisory node. When the transmission medium goes idle, nodes wait for their time slot in accordance with their assigned transmission order. The node assigned to transmit first transmits immediately. The node assigned to transmit second waits one time slot (but by that time the first node has already started transmitting). Nodes monitor the medium for transmissions from other nodes and update their assigned order with each detected transmission (i.e. they move one position closer to the front of the queue).

In order to access the underlying shared media, theoretically, the Ethernet protocol uses the CSMA/CD as Media Access Control (MAC) protocol, but since switches are now commonly used in industrial networks instead of hubs, CSMA/CD is not really used anymore.

2.2.1 Switched Ethernet

A switch is a hardware device that operates on the second layer of the OSI model. Switched networks use switches instead of hubs with a dedicated segment for each station. Since the only devices on the segments are the switch and the end station, the switch picks up every transmission before it reaches another node. The switch then forwards the frame over the appropriate segment that contains only a single node so the frame only reaches the intended recipient. For example, let us consider four devices A, B, C and D each one connected to one port of the switch. When A

communicates with B there is no interaction with the C and D's ports and a message exchange between them can occur separately from A and B. The switch has also an internal buffer so if C sends a message to A while A and B are speaking, the switch saves the message and forward it to A when the segment is free. Some switches today can support hundreds of dedicated segments. This allows many conversations to occur simultaneously on a switched network without any collision. There are two main types of switches: Store & forward and Cut-through. They differ in the way they forward incoming frames. A store & forward switch stores the whole frame and checks the checksum before forwarding it to its destination. This method improves reliability but the drawback is a longer transmission latency. A cut-through switch forwards the frame as soon as it is possible. The minimum requirement is that the switch knows the destination address. Naturally, cut-through switch has a very short frame-forwarding latency independent of the frame length but there is a risk of transmitting erroneous frames.

To aim a better level of determinism two properties of the switch are needed: full-duplex and VLAN priority.

2.2.2 Full-duplex

Nodes sharing a half-duplex connection are operating in the same collision domain (Figure 4). This means that these nodes will compete for bus access, and their frames can potentially collide with other frames on the network. Unless the access to the bus is controlled at a higher level and highly synchronized across all the nodes co-existing on the collision domain, collisions can occur and real-time communication is not guaranteed.

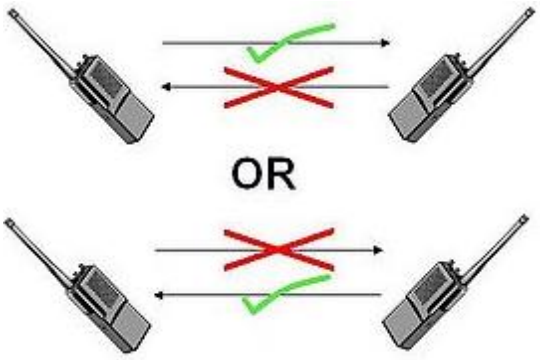


Figure 4 Half-duplex mode

With full-duplex, a node can transmit and receive simultaneously (Figure 5). A maximum of two nodes can be connected on a single full duplex link. Typically, this would be a node-to-switch or switch-to-switch configuration. Full-duplex communication provides every network node with a unique collision domain (Figure 6). This operation completely avoids collisions and does not even implement the traditional Ethernet CSMA/CD protocol.

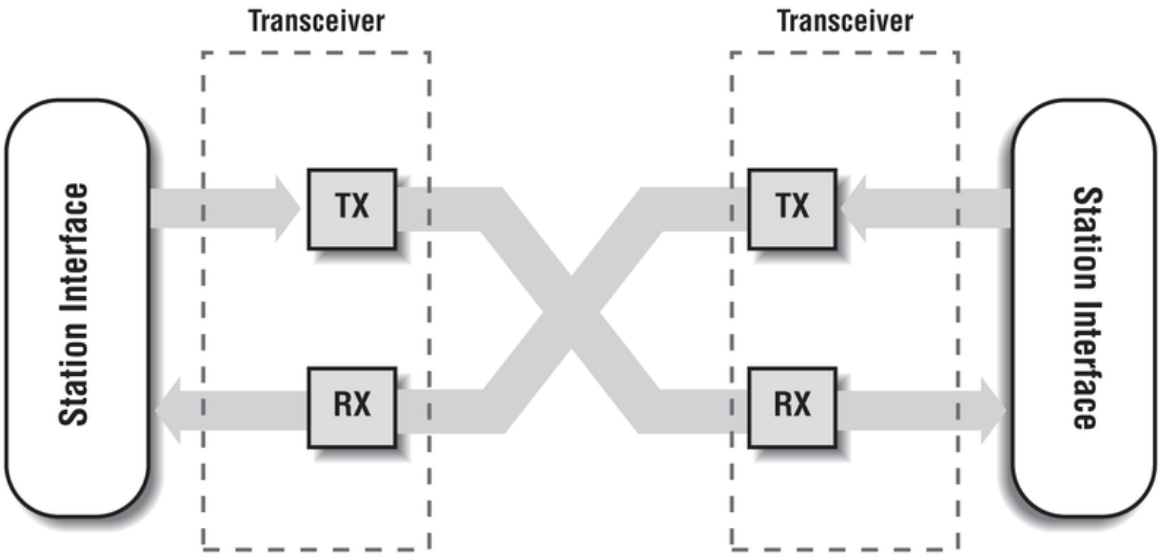


Figure 5 Full-duplex mode

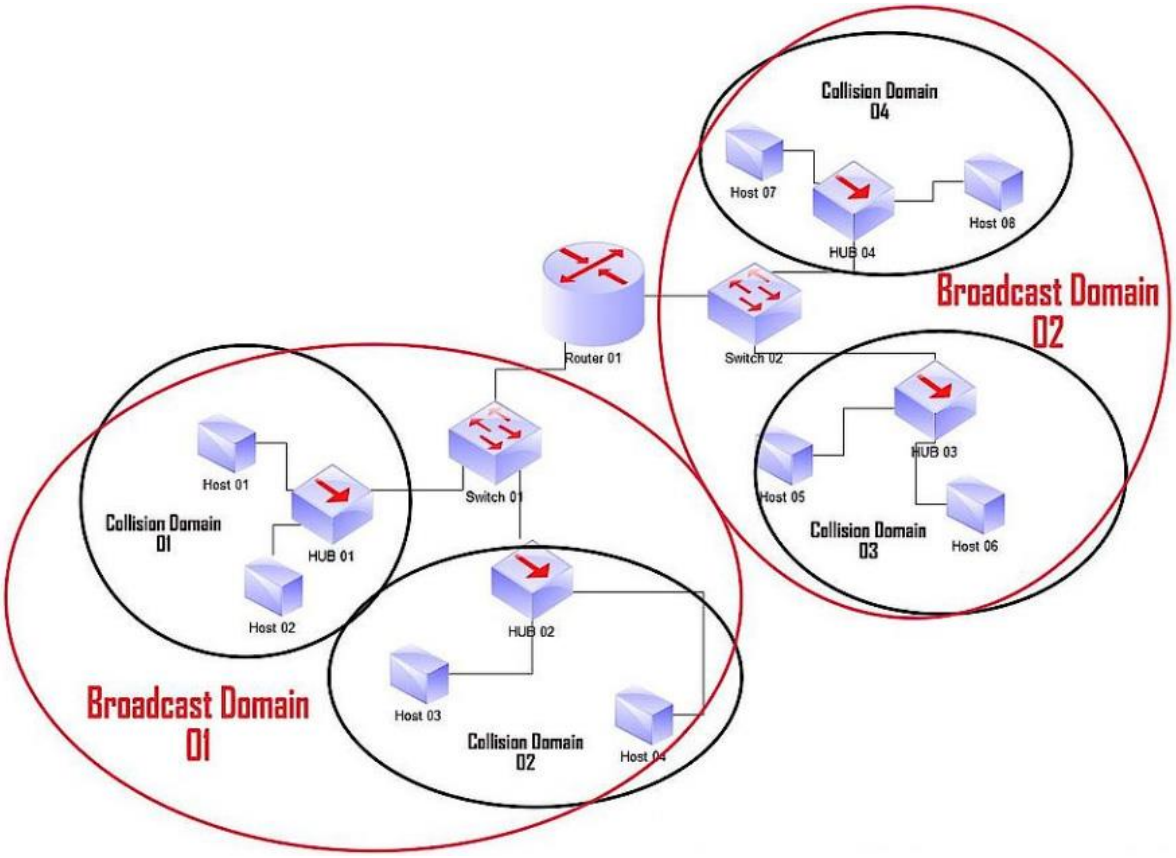


Figure 6 Example of different collision domains

2.2.3 Ethernet frame

A data packet on an Ethernet link is called an Ethernet packet, which transports an Ethernet frame as its payload. The Ethernet packet structure compliant with IEEE 802.1Q is shown in Figure 7.

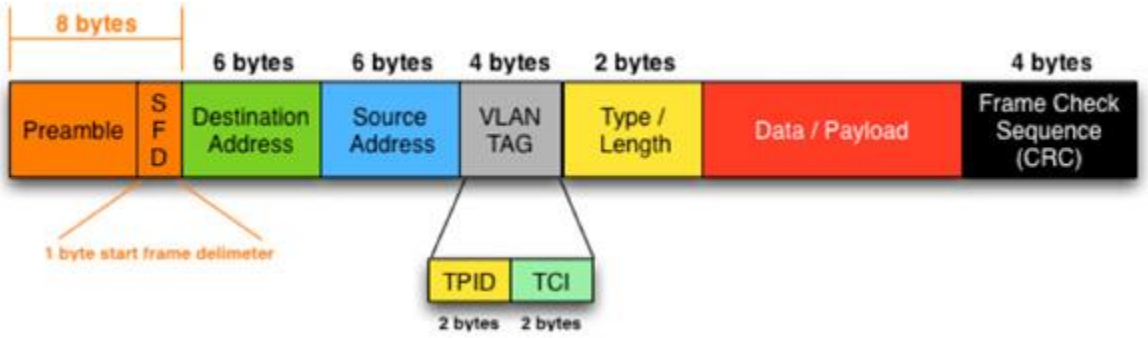


Figure 7 The Ethernet packet with IEEE802.1Q VLAN Tag

Below is an analysis of all the Ethernet parameters:

- Preamble: is seven bytes long, it consists of a pattern of alternating ones and zeros, and this informs the receiving stations that a frame is starting as well as enabling synchronisation
- Start of Frame Delimiter (SFD): consists of one byte and contains an alternating pattern of ones and zeros but ending in two ones
- Destination Address: contains a 48-bit long string indicating the Medium Access Control (MAC) address whose packet is destined.
- Source Address: contains a 48-bit long string indicating the source's Medium Access Control (MAC) address
- VLAN Tag: Virtual LANs are mainly used to form subnets in a logical LAN by using switches. IEEE802.1Q defines an additional 4 bytes VLAN-Tag that is inserted into the Ethernet frame (Figure 7). The VLAN-Tag field is inserted between the sender address and the length/type fields. It includes four additional fields:
    - Tag Protocol Identifier TPID is a 16-bit field set to a value of 0x8100 in order to identify the frame as an IEEE 802.1Q-tagged frame. This field is located at the same position as the EtherType/length field in untagged frames and is thus used to distinguish the frame from untagged frames. The original length/type field has its normal function.
    - Priority Code Point PCP a 3-bit field which refers to the IEEE 802.1p class of service and maps to the frame priority level. Values in order of priority goes from 1 (background), to 7 (network control). These values can be used to prioritize different classes of traffic (voice, video, data, etc.).
    - Canonical Format Indicator CFI it is a 1-bit field. It may be used separately or in conjunction with PCP to indicate frames eligible to be dropped in the presence of congestion.
    - VLAN Identifier VID a 12-bit field specifying the VLAN to which the frame belongs. The hexadecimal values of 0x000 and 0xFFF are reserved. All other values may be used as VLAN identifiers, allowing up to 4094 VLANs.

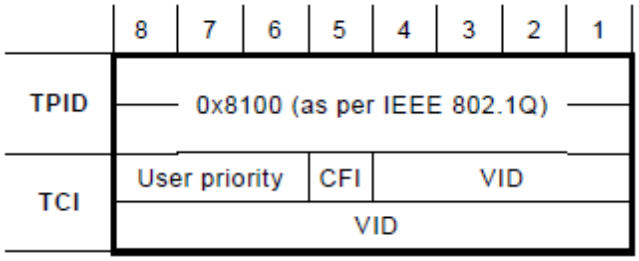


Figure 8 VLAN Tag for IEEE802.1Q Ethernet Frame

- EtherType: it is a two-octet field in an Ethernet frame. It is used to indicate which protocol is encapsulated in the payload of the frame such that the Ethernet level knows to which application layer forward the frame.
- Data/Payload: This block contains the payload data and it may be up to 1500 bytes long. The minimum payload is 42 octets when an 802.1Q tag is present and 46 octets when absent. If the length of the field is less than 42/46 bytes, then padding data is added to bring its length up to the required minimum of 42/46 bytes.
- Frame Check Sequence (FCS): is a four-octet Cyclic Redundancy Check (CRC) that allows detection of corrupted data within the entire frame as received on the receiver side.

  A CRC-enabled device calculates a short, fixed-length binary sequence, known as the check value or CRC, for each block of data to be sent or stored and appends it to the data, forming a code-word. When a code-word is received or read, the device either compares its check value with one freshly calculated from the data block, or equivalently, performs a CRC on the whole code-word and compares the resulting check value with an expected residue constant. If the CRC check values do not match, then the block contains a data error. FCS function is further analysed in section 2.2.5.

2.2.4 Fast Ethernet

It consists in a series of standards that takes the speed of Ethernet from 10 Mbps to 100 Mbps. While this is not a great benefit in process control applications where the dimension of data is small, it significantly reduces the back-off time introduced by a collision. The reason of the use of 100 Mbps Ethernet in the industrial world is due to

the trend of using the same physical bus for heterogeneous traffic and not only for process control.

The question of whether Ethernet will find its way down to the factory-floor became closely linked to the real-time issue. Currently many automation producers and different alliances have developed their own real-time Industrial Ethernet standards. As a result, there are now many real-time Industrial Ethernet solutions on the market that are each best suited to a particular application field but also incompatible with each other. Almost all the solutions use 100Mbps full-duplex, fully switched Ethernet but that is where the consistency usually ends. Some of the solutions restrict the amount of other traffic e.g. TCP/IP in the network or in the segments where real-time process data is carried. Thus, they are not even close to Ethernet standard anymore.

Although such an Ethernet network can be considered collision-free, packets may still experience corruption during their transmission due to a fault or interference.

2.2.5 Ethernet FCS

All frames, bits, bytes, and fields contained within the Ethernet packet, are susceptible to errors from a variety of sources. In this regard, inside the Ethernet PDU a specific hash function for error detecting, called Frame Check Sequence (FCS), containing a 4-octet (32-bit) value, can be highlighted. The FCS field contains a number that is calculated by the source node based on the data in the frame. This number is added to the end of a frame that is sent. When the destination node receives the frame, it recalculates the FCS number and compares with FCS number included in the frame. If the two numbers are different, an error is assumed and the frame is discarded.

The FCS is often transmitted in such a way that the receiver can compute a running sum over the entire frame, together with the trailing FCS, expecting to see a fixed result (such as zero) when it is correct. For Ethernet and other IEEE802 protocols, this fixed result, also known as the magic number or CRC32 residue, is equal to 0xC704DD7B. When transmitted and used in this way, FCS generally appears immediately before the frame-ending delimiter.

A cyclic redundancy check (CRC) is used by the transmitting algorithms to generate a CRC value for the FCS field. This value is computed as a function of the contents of the protected fields of the MAC frame: the Destination Address, Source Address, Length/ Type field, MAC Client Data, and Pad (that is, all fields except FCS). The encoding is defined by the following generating polynomial.

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Mathematically, the CRC value corresponding to a given MAC frame is defined by the following procedure:

- The first 32 bits of the frame are complemented
- The n bits of the protected fields are then considered to be the coefficients of a polynomial M(x) of degree n-1. (The first bit of the Destination Address field corresponds to the $x^{(n-1)}$ term and the last bit of the MAC Client Data field (or Pad field if present) corresponds to the x^0 term.)
- M(x) is multiplied by x^{32} and divided by the G(x) defined above, producing a remainder R(x) of degree ≤ 31
- The coefficients of R(x) are a 32-bit sequence
- The bit sequence is complemented and the result is the CRC.

The 32 bits of the CRC value are placed in the FCS field so that the x^{31} term is the left-most bit of the first octet, and the x^0 term is the right most bit of the last octet. (The bits of the CRC are thus transmitted in the order $x^{31}, x^{30}, \ldots, x^1, x^0$).

To conclude, if an Ethernet message experiences corruption during its transmission, the receiving device can detect such an issue through the Ethernet FCS frame, therefore the affected packet can be identified and discarded.

2.3 PROCESS CONTROL: PROFINET

Profinet is an industrial protocol based on an Ethernet infrastructure and it has been developed by Profibus International (PI). Its main goals are the significant reduction of project and installation costs and a full integration between field and control room. With Profinet is possible to use both a Real Time communication for time critical tasks than a slower data exchange based on TCP/IP for not Real Time applications. It also merges the ideas of the new automation based on Industrial Ethernet and the old one based on fieldbus, like Profibus. Profinet is defined by the standard IEC 61784-1.

The PROFINET concept has two perspectives: Profinet CBA and Profinet IO. PROFINET CBA is suitable for component-based machine-to-machine communication via TCP/IP and for real-time communication required for modular plant designs. It enables a simple modular design of plants and production lines based on distributed intelligence using graphics-based configuration of communication between intelligent modules. Recently, the complex PROFINET CBA has been supplemented by the much simpler PROFINET IO standard. Contrary to CBA, PROFINET IO is targeted at decentral periphery scenarios. Analogous to PROFIBUS DP, it is based on a modular device model and relies on the cyclic exchange of messages between device and supervisory system, typically a PLC. Indeed, PROFINET IO is based on 15 years of experience with the successful Profibus DP and combines the normal user operations with the simultaneous use of innovative concepts of the Ethernet technology. This ensures the smooth migration of PROFIBUS DP into the PROFINET world. Using switched Ethernet, this protocol achieves the highest performance in terms of time response. Profinet IO supports all types of communication (no-RT, Real Time (RT) and Isochronous Real Time (IRT)) and is intended for hard RT applications. PNIO is typically used for process control in the area of motion control and machine control, and to create a high-speed backbone for process control[3].

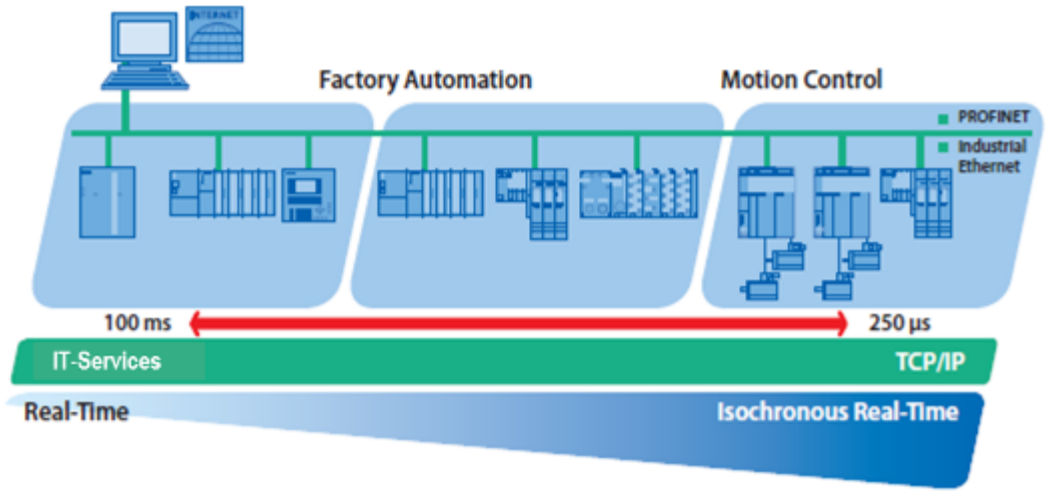


Figure 9 Profinet is suitable for all industrial applications

2.3.1 Profinet OSI model

According to IEC 61158, Profinet, as a fieldbus, is defined in several levels, or layers, following the ISO/OSI model. These layers characterize the internal functions of a communication system by partitioning it into abstraction layers, ordered in a hierarchical disposition. A layer serves the layer above it and is served by the layer below it. Industrial protocols are mainly defined by the layers 1, 2 and 7 but Profinet uses also the layers 3 and 4 to provide non-Real Time data exchange.

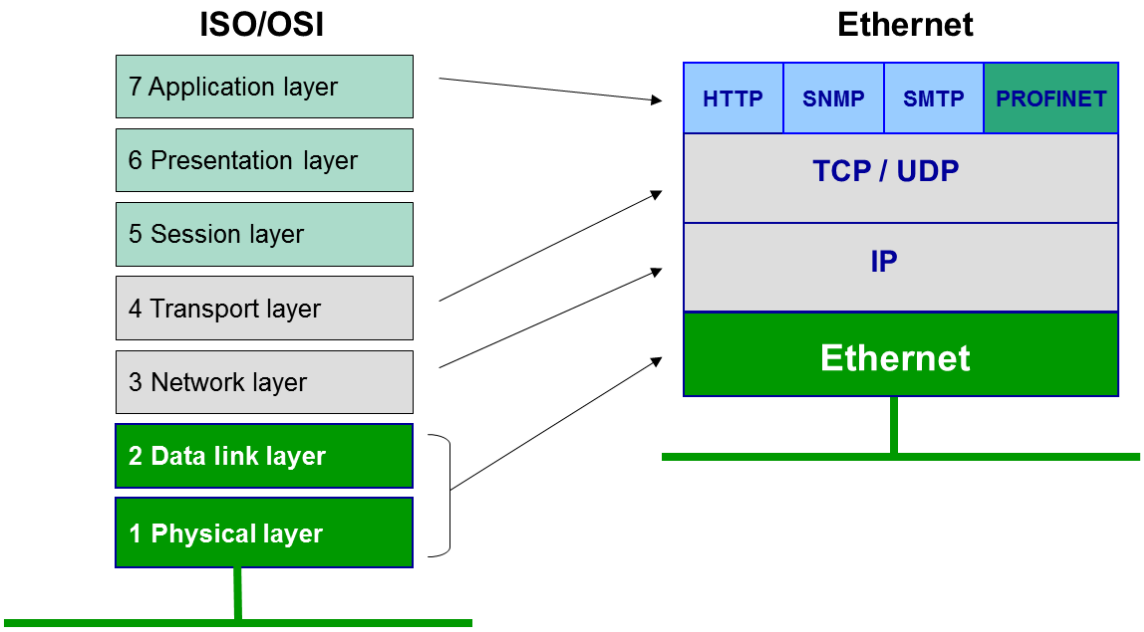


Figure 10 Profinet OSI model

The first level is the Physical Layer that defines the electrical and physical specifications of the data connection such as the layout of pins, voltages and cable specifications. Profinet can run on copper or fiber optic, using 100Base-TX and 100Base-FX standards.

Data Link Layer is the second level of the OSI model and defines the structure of the frame and the medium access control mechanism. The technology used by Profinet is Ethernet, but the use of switches is mandatory to control collisions on the bus. On the third and fourth layers Profinet uses an UDP/IP stack. These 2 layers are used only when the communication requires more than 100ms, for setup, configuration and maintenance functions.

Profinet IO can manage different types of traffic, divided in the following classes:

- No-RT or RT_CLASS_UDP: it is used for the data transmission that requires more than 100ms with high level of jitter. It makes relevant use of UDP/IP protocols for setup, configuration and maintenance functions:
    - DHCP - Dynamic Host Configuration Protocol
    - SNMP - Simple Network Management Protocol

- ARP – Address Resolution Protocol.

This channel is used for non-time critical tasks such as:

- Downloading of configuration parameters
- Diagnostics
- Device management information.

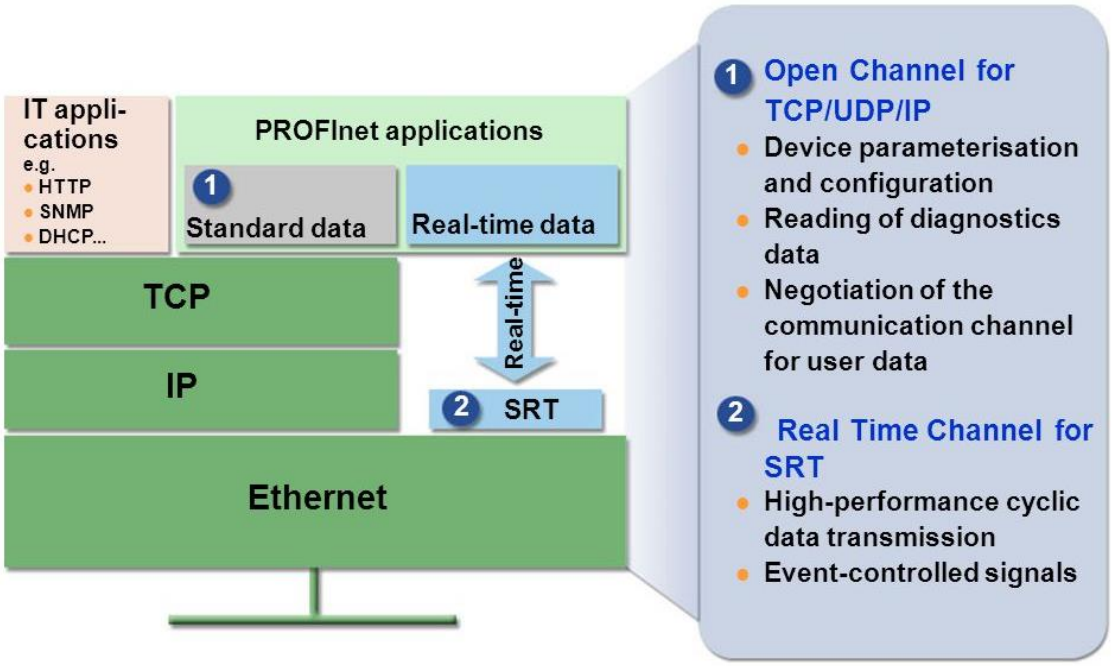


Figure 11 Profinet communication channels

- RT_CLASS_1: In order to achieve a cycle time in the order of a few milliseconds with moderate jitter in soft real time scenarios, the RT communication bypasses the UDP/IP stack and directly accesses the Ethernet MAC layer. RT frames are identified by the value 0x8892 in the Ethertype field. The Real-Time channel is used for time-critical data such as: cyclic process data, alarms and critical messages, communication monitoring.
- RT_CLASS_2 or IRT Flex: to achieve cycle times of 1ms a synchronized communication is required. IRT employs the “Precision Transparent Clock Protocol” (PTCP) to synchronize the clocks in all stations and all switches. This allows the scheduling of a precise cycle start also in large distributed systems with many switches. IRT requires hardware support by ASICs in all station as well as special switches.

- RT_CLASS_3 or IRT Top: it is the most powerful class of Profinet and it fits the requirements for motion control applications. It acts like RT_CLASS_2 but it needs also information about the topology of the network in order to determine the delay of each segment.

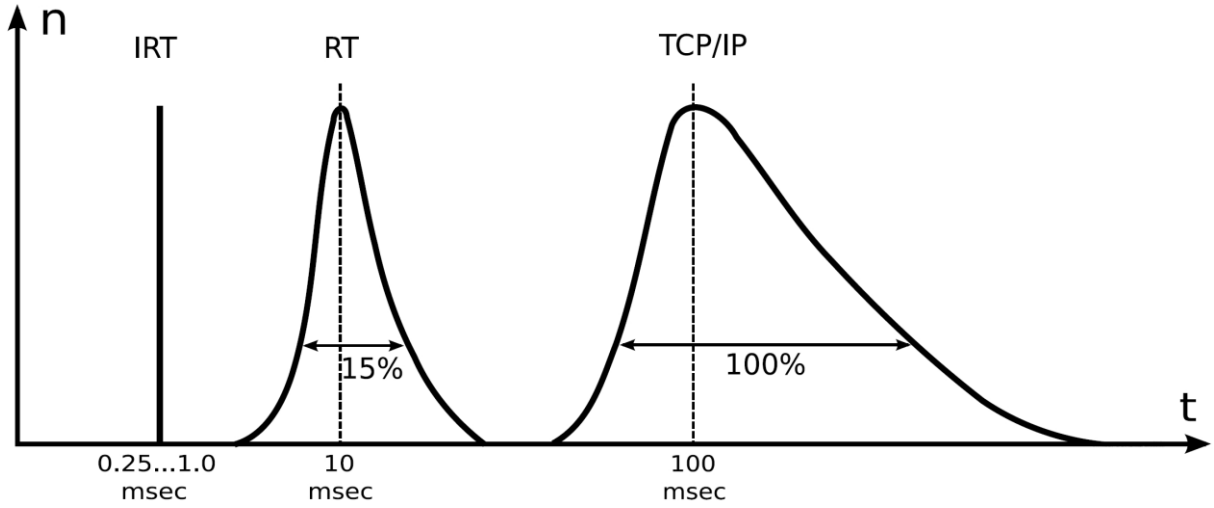


Figure 12 Profinet channels performances

2.3.2 Profinet system devices

PROFINET IO aims at application scenarios, where a central station communicates with decentral field devices, like what happen with Profibus. With Profinet IO the medium access control has changed from Master/Slave to Producer/Consumer.

Master/Slave is a model where one device control and manage the bus access of one or more other devices, called slaves, those can use the bus only after the authorization of the master device. The Master/Slave model is unable to satisfy the trend of automation and decentralize the tasks of the automation (distributed automation), since if the master fails all the functions connected to it fails too. The Producer/Consumer model suits better the needs of distributed automation. Every node can take the control of the network and be temporarily the Producer of data; in this moment all the other devices act as Consumers of that certain data. The producer doesn't have to specify a destination address but every node knows both the variables it needs and the variables it produces.

Each station in a Profinet IO system can take one of the following roles (Figure 13):

- IO – Controller: the IO-Controller represents an intelligent central station, typically a programmable logic controller (PLC) on which the automation program runs. This is comparable to a class 1 master in Profibus. It is responsible for the configuration or parameterization of its associated devices. The IO-Controller provides output data to the configured IO-Devices in its role as provider and is the consumer of input data of IO-Devices.
- IO – Device: an IO-Device is a distributed I/O field device, like an analog input unit, that is connected to one or more IO-Controllers via Profinet IO. It is comparable to the function of a slave in Profibus. It cyclically transmits collected process data to the IO-Controller and vice versa. It also provides diagnostic or alarm information to the IO-Controller.
- IO – Supervisor: this can be a programming device, personal computer (PC), or human machine interface (HMI) device for commissioning or diagnostic purposes and corresponds to a class 2 master in Profibus.

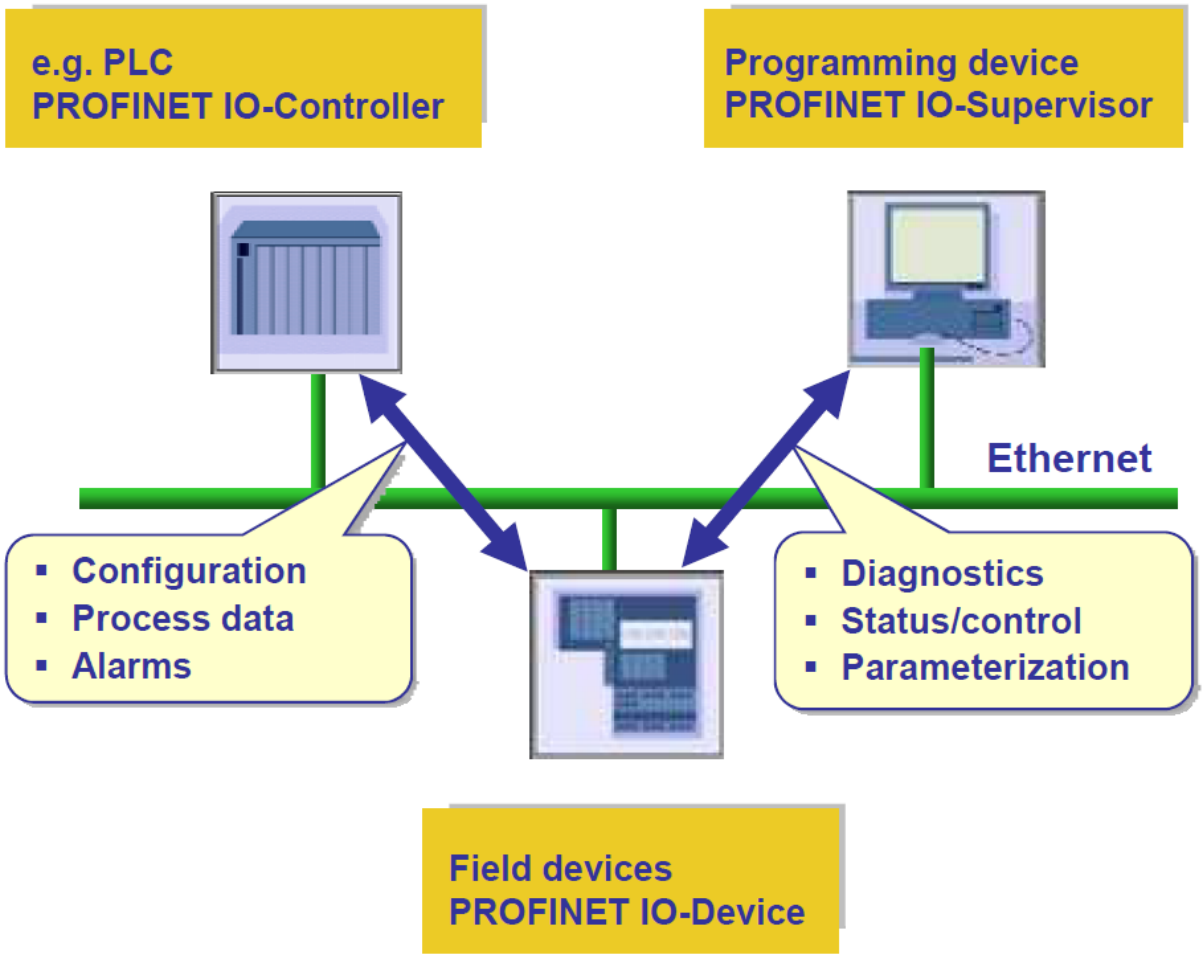


Figure 13 Communication paths for Profinet IO

A plant unit has at least one IO-Controller and one or more IO-Devices. IO-Supervisors are usually integrated only temporarily for commissioning or troubleshooting purposes.

2.3.3 Model of an IO-Device

A unique model has been defined for IO-Devices that allows the configuration of every module of the device. This model inherits many of its features by the one developed for Profibus DP. Profinet differentiates between compact field devices, in which the degree of expansion is already specified in the as-delivered condition and cannot be changed by the user, and modular field devices, in which the degree of expansion can be customized for a specific application when the system is configured. The device model describes all field devices in terms of their possible technical and functional features. It is specified by the DAP (Device Access Point) and the defined modules for

a particular device family. A DAP is the access point for communication with the Ethernet interface and the processing program. A variety of I/O modules can be assigned to it in order to manage the actual process data communication. The following structures are standardized for an IO-Device:

- The slot designates the place where an I/O module is inserted in a modular I/O field device. The configured modules containing one or more subslots for data exchange are addressed based on the different slots.
- Within a slot, the subslots represent the actual interface to the process (inputs/outputs). The granularity of a subslot (bitwise, bytewise, or wordwise division of I/O data) is determined by the manufacturer. The data content of a subslot is always accompanied by status information, from which the validity of the data can be derived.
- The index specifies the data within a slot/subslot that can be read or written acyclically via read/write services. For example, parameters can be written to a module or manufacturer-specific module data can be read out based on an index. Cyclic I/O data are addressed by specifying the slot/subslot combination. These can be freely defined by the manufacturer. For acyclic data communication via read/write services, an application can specify the data to be addressed using slot, subslot, and index.

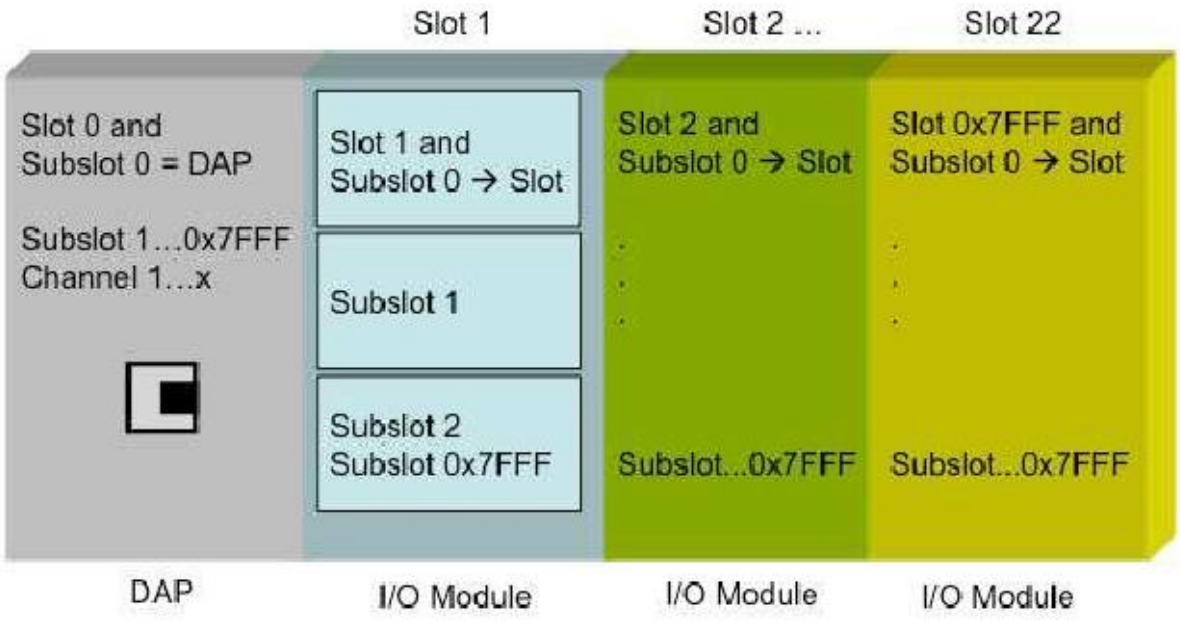


Figure 14 IO-Device model

To configure a Profinet network a description of all the devices that form the network is needed. In Profibus the GSD (General Station Description) file contains this information; the file that is used in Profinet takes the name of GSDML because it's written in XML format. This file can be read with a text editor or with a common web browser.

2.3.4 Engineering, addressing and communication relations of an IO system

Each IO-Controller manufacturer provides an engineering tool for configuring a PROFINET system. During system engineering, the configuring engineer joins together the modules/submodules of an IO-Device defined in the GSDML file in order to map them to the real system and to assign them to slots/subslots. The configuring engineer configures the real system symbolically in the engineering tool. Figure 15 shows the relationship between the GSD definitions, configuration, and real plant view. After completion of system engineering, the configuring engineer downloads the system data to the IO-Controller, which also contains the system specific application. As a result, an IO-Controller has all the information needed for addressing the IO-Devices and for data exchange. Before an IO-Controller can perform data exchange

with the IO-Devices, these must be assigned an IP address based on their configured name. This must take place prior to system power-up.

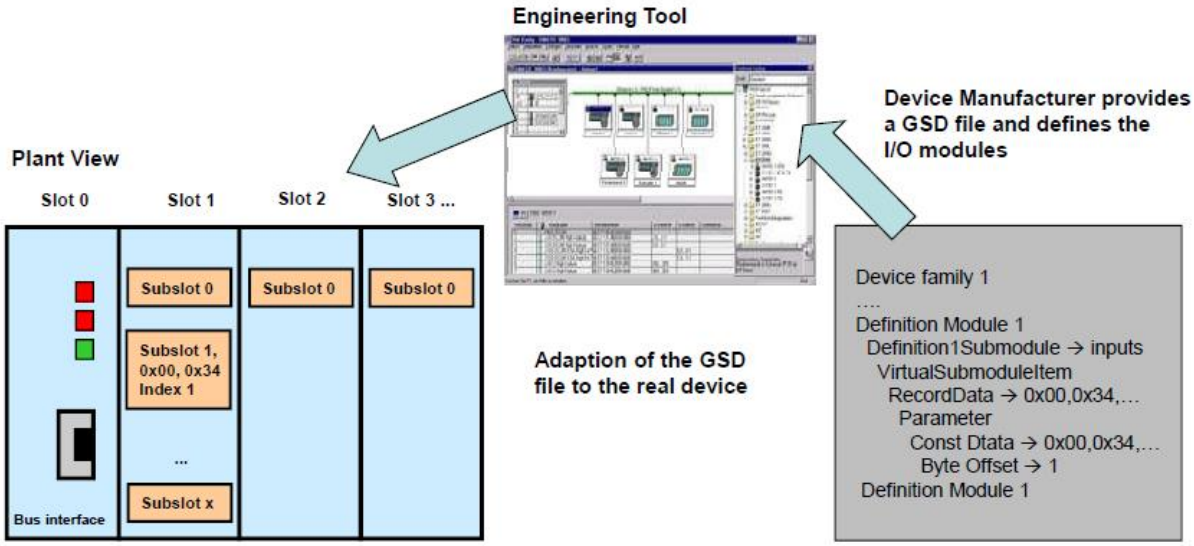


Figure 15 Adaption of the GSDML file to the real device

In PROFINET IO, each field device has a symbolic name that uniquely identifies the field device within a PROFINET IO system. This name is used for assigning the IP address. The DCP protocol (Dynamic Configuration Protocol) integrated in every IO-Device is used for this purpose. The IP address is assigned with the DCP protocol based on the device name. Because DHCP (Dynamic Host Configuration Protocol) is in widespread use internationally, PROFINET has provided for optional address setting via DHCP or via manufacturer-specific mechanisms. The addressing options supported by a field device are defined in the GSDML file for the respective field device. Optionally, the name can also be automatically assigned to the IO-Device by means of a specified topology based on neighbourhood detection.

To establish communication between the higher-level controller and an IO-Device, the communication paths must be established. These are set up by the IO-Controller during system start up based on the configuration data in the engineering system. Every data exchange is embedded into an AR (Application Relation) (Figure 16). Within the AR, CRs (Communication Relations) specify the data explicitly. As a result, all data for the device modelling, including the general communication parameters, are downloaded to the IO-Device. An IO-Device can have multiple ARs established from

different IO-Controllers. The IO-Controller performs this automatically using the DCP protocol. After a start up/restart, an IO-Controller always initiates system power-up based on the configuration data without any intervention by the user. During system power-up, an IO-Controller establishes an explicitly specified communication relation (CR) and application relation (AR) with an IO-Device. This specifies the cyclic I/O data, the alarms, the exchange of acyclic read/write services, and the expected modules/submodules. After successful system power-up, the exchange of cyclic process data, alarms, and acyclic data traffic can occur.

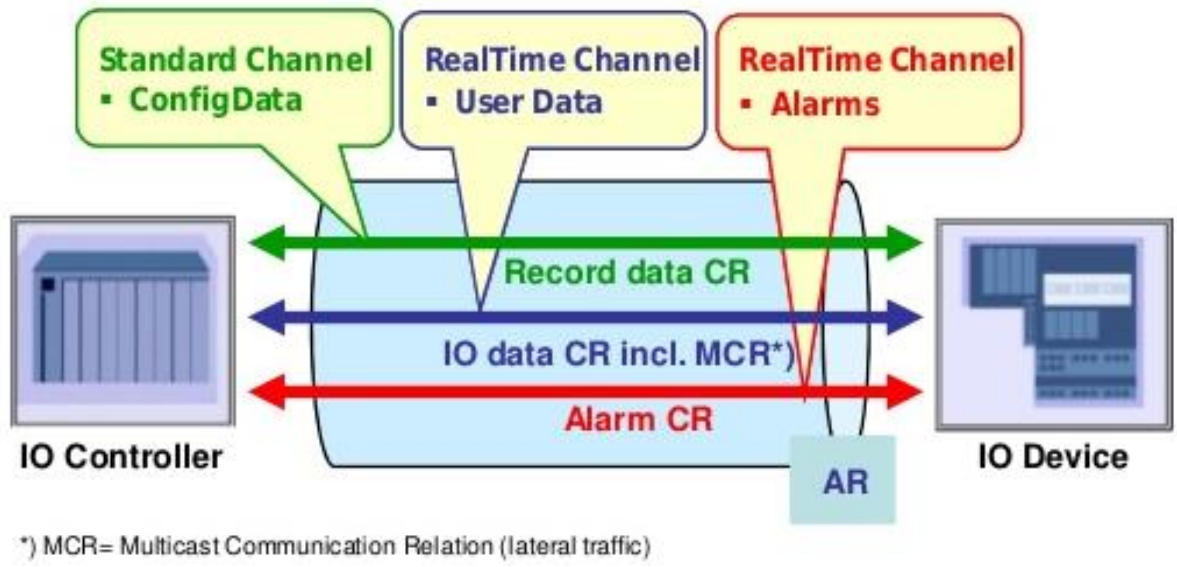


Figure 16 Application relation and Communication relations

2.3.5 Profinet RT cycle

The Soft Real Time Channel, or RT channel, is the first step of the real time communication and it is compliant for process in which a delay of about 10 ms can be accepted.

This time specification identifies the typical area of application of the RT communication as the factory automation. To achieve this, the communication stack consists only on the first and second levels and the application, bypassing TCP/IP or UDP/IP of the NRT communication.

Figure 17 presents the structure of the Profinet RT frame.

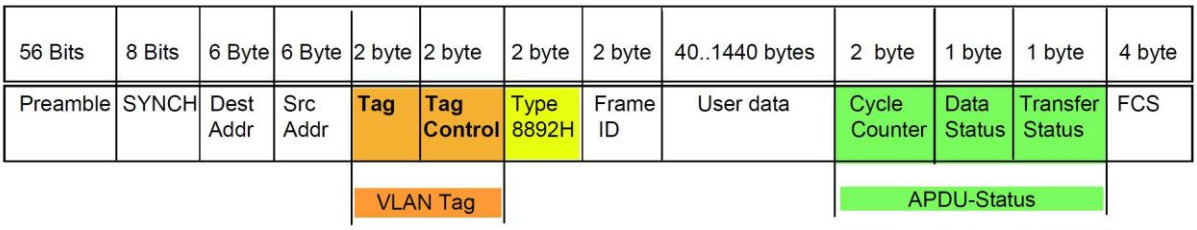


Figure 17 Profinet RT frame

It consists of:

- Preamble: sequence of bit used to synchronize the communication;
- SFD: Start Frame Delimiter is the sequence that indicates the beginning of the frame;
- Destination Address: MAC address of the destination node;
- Source Address: MAC address of the source node;
- EtherType: 0x8100 specify the VLAN header;
- VLAN: contains the specification of the VLAN (Profinet RT requires priority level 6);
- EtherType: 0x8892 for Profinet RT;
- Frame ID: indicates the type of the frame;
- Data;
- Cycle Counter: a bit represents an increment of 31,25 µs;
- Data Status
- Transfer Status;
- FCS: Frame Control Sequence is used to verify if the frame transmitted is correct.

FrameID	Meaning
0x0000 – 0x00FF	Time Synchronization
0x0100 – 0x7FFF	RT_CLASS_3 Frames (IRTtop)
0x8000 – 0xBFFF	RT_CLASS_2 Frames (IRTflex)
0xC000 – 0xFBFF	RT_CLASS_1 Frames (RT/UDP)
0xFC00 – 0xFCFF	Acyclic transmission „high“
0xFD00 – 0xFDFF	Reserved
0xFE00 – 0xFEFC	Acyclic transmission „low“
0xFEFD – 0xFEFF	DCP
0xFF00 – 0xFFFF	reserved

Figure 18 Profinet Frame Ids

The Frame ID allows the destination node to identify the received frame in order to quickly classify it in the right communication channel. The Cycle Counter is written in the frame by the provider of the message. When the consumer receives the frame, it verifies the value of the counter to control if the transmitted data are “up to date”.

2.4 POWER AUTOMATION: IEC 61850

2.4.1 Overview

Over the last decade, the “digitization” of the electron enterprise has grown at exponential rate. Utility, industrial, commercial, and even residential consumers are transforming all aspects of their lives into the digital domain. Moving forward, it is expected that every piece of equipment, every receptacle, every switch, and even every light bulb will possess some type of setting, monitoring and/or control.

Electrical devices were based on electromechanical circuits and the automation functions were actuated by the control room. The increasingly widespread use of microprocessor technology changed this paradigm; nowadays these devices are considered the main actors of the automation and the protection in a substation and they earned the name of IED (Intelligent Electrical Device). This shift also makes necessary the change from analogic communication to digital data exchange, from copper wires to a communication standard on fieldbus.

An Intelligent Electrical Device must be built respecting all the specifications about electromagnetic compatibility and, especially for the components that provide critical functions, it must be compliant for high levels of reliability and availability thanks to the installation of redundancy components. Furthermore, protective functions are designed in order to make the IED independent when it must act on a fault: the protection process involves only the sensors (current transformers and voltage transformers), the electrical device and the actuator (circuit breaker). The upper levels of the automation don’t provide any commands and they just receive the information about the operation actuated by the IED.

CPUs significantly increase the functions that an electrical device can perform. Some examples are:

- Real-time Measures: IED can elaborate simple data from the sensors and provide more complex measures;

- Event log: it is possible to record and store data or events acquired by the sensors connected to the IED. For example, faults, trips, exceeded thresholds and interlocks;
- Complex elaborations: they can be real-time or not. For example, the localization of the fault for a distance protection (real-time) or frequency analysis, phasor diagrams and oscilloperturbography;
- Time tagging: events are provided with a time stamp that indicates the time at which the event occurred with a precision of 1 ms;
- Several protection functions: while old electromechanic relays can perform only one protective function, a smart electrical device can perform a lot of them by configuring different logics. For example, one IED can perform overcurrent protection, distance protection, current direction protection, over and under voltage protection and more;
- Fast communication between IEDs: some devices can exchange data in fast way to satisfy critical tasks such as interlocks, backups or selectivity;
- Software configuration: the configuration of the relay can be made via engineering software in order to set different calibrations.

It easy to understand that only a fieldbus communication can sustain the big amount of data needed to satisfy these functions. IEC 61850 defines a communication model that fulfil these requirements.

IEC 61850 aims to provide interoperability between Intelligent Electronic Devices (IEDs) for protection, monitoring, metering, control and automation in substations. Interoperability and free allocation of functions open a vast range of possible solutions, but the consideration of customer requirements and commercially available equipment scales down this range to a handful of them. It is important for both utilities and substation automation system providers to understand this process. This design process will be illustrated in the following pages.

The basic functionality of Substation Automation (SA) is given by its tasks and will not be changed by IEC 61850. On a first look, also the system architecture is not so much changed. Nevertheless, communication is the backbone of SA, therefore IEC 61850 is

the most important key for designing systems. A lot of inherent features in IEC 61850, like the use of object-oriented data model and the selection of mainstream communication technology, allow responding very dedicated to requirements stated in customer specifications not by chance but based on standardized rules. For specification, design and engineering, the most important feature of IEC 61850 is its support to strong formal description of the substation. The use of this strong description facility will be further discussed later.

The first edition of the standard was primarily related to protection, control and monitoring. From 2009 onwards the original parts of the IEC 61850 series have been updated and extended to cover also measurement (including statistical and historical data handling) and power quality. The concepts defined in IEC 61850 have been applied beyond the substation domain (Figure 19):

- The modelling of hydropower plants (see IEC 61850-7-410) distributed energy resources
- The modelling of wind turbines has been standardized according to IEC 61850
- The communication has also been extended to substation communication (see IEC 61850-90-1).

The stated scope of IEC 61850 is to encourage communication and interoperability within the substation. “Interoperability” is the characteristic of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, in either implementation or access, without any restrictions. Interchangeability is not guaranteed, in fact, to be able to change a device from a manufacturer with an equivalent one from a different maker, further configuration operations are needed. IEC 61850 defines the various aspects of the substation communication network in 10 major sections as shown in the table below (Figure 20).

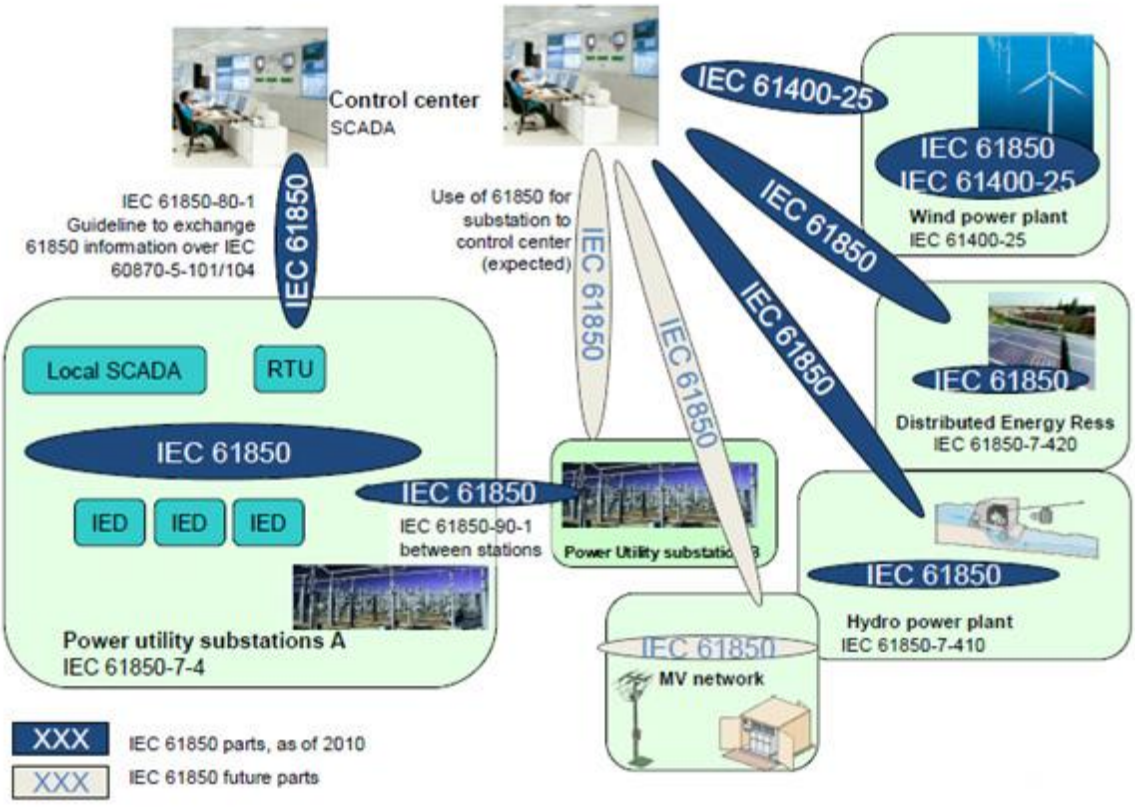


Figure 19 Scope of IEC 61850

Part	Title
1	Introduction and Overview
2	Glossary of terms
3	General Requirements
4	System and Project Management
5	Communication Requirements for Functions and Device Models
6	Configuration Description Language for Communication in Electrical Substations Related to IEDs
7	Basic Communication Structure for Substation and Feeder Equipment
7.1	Principles and Models
7.2	Abstract Communication Service Interface (ACSI)
7.3	Common Data Classes (CDC)
7.4	Compatible logical node classes and data classes
8	Specific Communication Service Mapping (SCSM)
8.1	Mappings to MMS (ISO/IEC 9506 – Part 1 and Part 2) and to ISO/IEC 8802-3
9	Specific Communication Service Mapping (SCSM)
9.1	Sampled Values over Serial Unidirectional Multidrop Point to-Point Link
9.2	Sampled Values over ISO/IEC 8802-3
10	Conformance Testing

Figure 20 IEC 61850 parts

Old generation substation automation communication protocols typically define how bytes are transmitted on the wire. However, they did not specify how data should be organized in devices in terms of the application. This approach requires power system engineers to manually configure objects and map them to power system variables and low-level register numbers, index numbers, I/O modules, etc. In this regards, IEC 61850 is unique. In addition to the specification of the protocol elements (how bytes are transmitted on the wire), IEC 61850 provides a comprehensive model for how power system devices should organize data in a manner that is consistent across all types and brands of devices. This eliminates much of the non-power system configuration effort because the devices can configure themselves. For instance, if you put a CT/PT input into an IEC 61850 relay, the relay can detect this module and automatically assign it to a measurement unit without user interaction.

The approach of the IEC 61850 series is to blend the strengths of the following three methods:

- Functional decomposition is used to understand the logical relationship between components of a distributed function, and it is presented in terms of logical nodes (LNs) that describe the functions, sub-functions and functional interfaces.
- Data flow modelling is used to understand the communication interfaces that must support the exchange of information between distributed functional components and the functional performance requirements.
- Information modelling is used to define the abstract syntax and semantics of the information exchanged and is presented in terms of data object classes and types, attributes, abstract object methods (services), and their relationships.

In order to cope with the fast innovation of communication technology IEC 61850 makes the communication independent from the application by specifying a set of abstract services and objects. In this way, applications can be written in a manner that is independent from a specific protocol. This abstraction allows both vendors and utilities to maintain application functionality and to optimize this functionality when appropriate (Figure 21). It also allows, as the scope of IEC 61850 is wider and wider, to cope with the diversity of communication solutions required by the new targeted domains, while keeping the same data model[25].

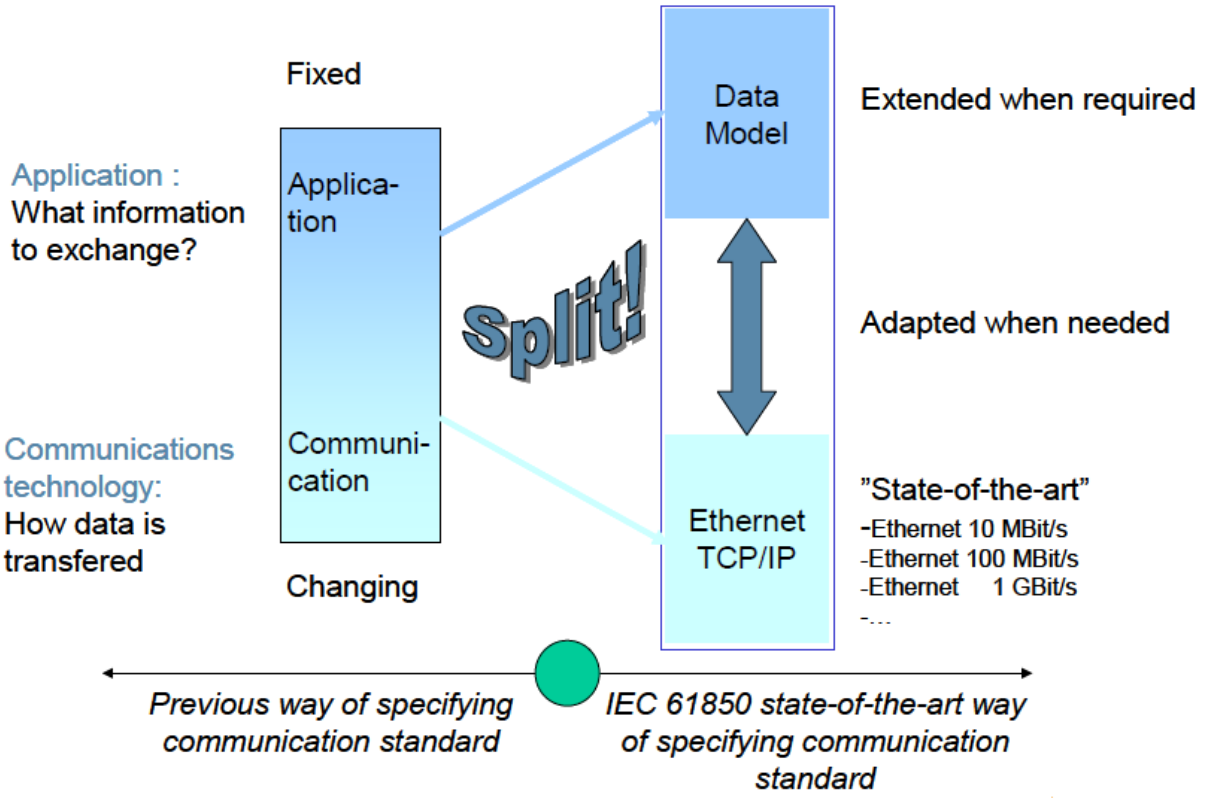


Figure 21 How IEC 61850 copes with fast innovation of communication technology

Besides the new communication model described above, IEC 61850 introduce the new concepts of Station Bus (IEC 61850-8-1) and Process Bus (IEC 61850-9-2), as well as their communication techniques.

According to Figure 22, a substation control system can be divided into three distinct levels:

- Station Level
- Bay/Unit Level
- Process Level

Station level devices usually consist of the station computer with a database, the operator's workplace and interfaces for remote communication. Thus, the station level includes the overall substation-wide coordination, substation Human Machine Interface (HMI), and the SCADA system, while protective functions are to be performed at the Bay Level.

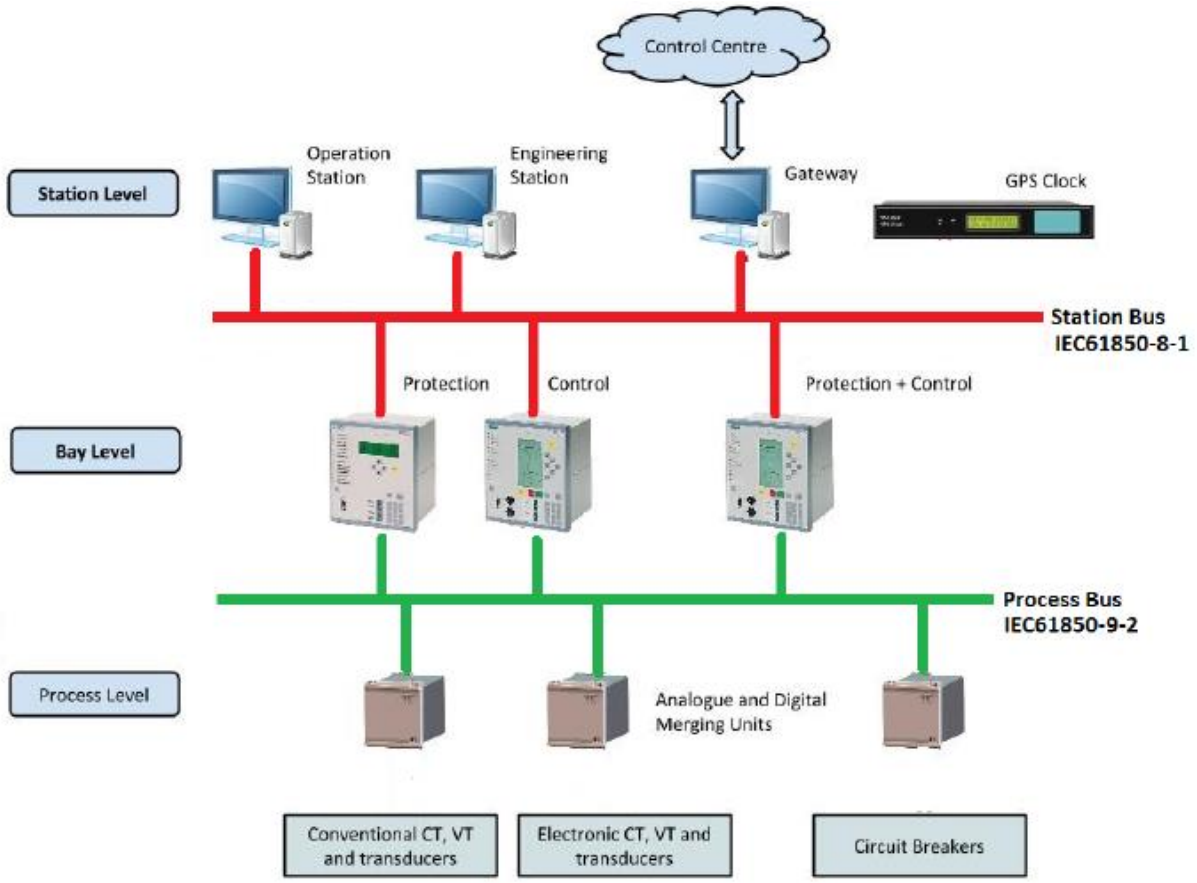


Figure 22 Substation levels in IEC 61850

As technology migrates to “next generation” low-energy voltage and current sensors, the ability to digitize the base quantities at the source and transmit the resulting sample values back to the substation becomes a need. In addition to Sampled Values, the ability to remotely acquire status information as well as set output controls is very desirable. IEC 61850 addresses this need through the definition of Sampled Measured Values (SMV) services and the implementation of a Process Bus. The Process level of the substation is related to collect information, such as Voltage, Current, and status information, from the transformers and transducers connected to the primary power system process.

Figure 23 shows the basic concept of the Process Bus. This approach implies that instrument transformers need to be equipped with built-in Analog-to-Digital Converters (ADC) and appropriate data formatting capability required for generating the SV messages. Instrument transformers that do not have this capability (for example,

conventional CTs and PTs) would be connected to yard mounted Merging Units intended to bridge the gap between the analog signal world and the IEC 61850 process bus LAN.

The Merging Units sample the signals at an agreed, synchronized rate. In this manner, any IED can input data from multiple MUs and automatically align and process the data. Currently, there is an implementation agreement that defines a base sample rate of 80 samples per power system cycle for basic protection and monitoring and a “high” rate of 256 samples per power system cycle for high-frequency applications such as power quality and high-resolution oscillography.

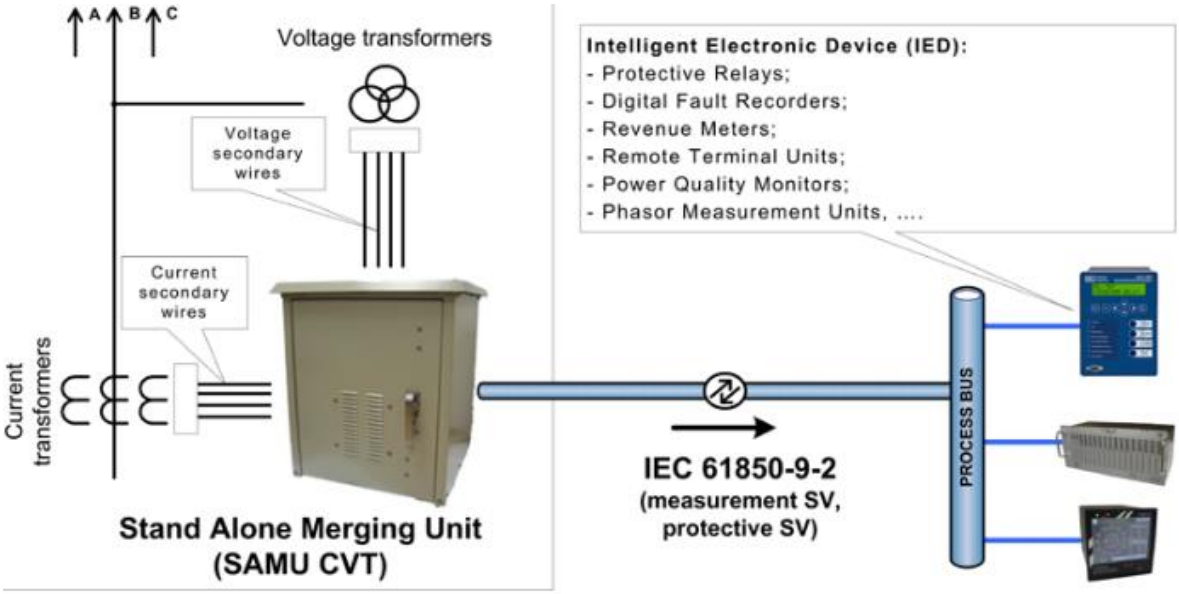


Figure 23 Merging Unit used in Process Bus

2.4.2 Modelling

The information exchange mechanisms rely primarily on well-defined information models. These information models and the modelling methods are the core of the IEC 61850 series. All information made available to be exchanged with other devices is defined in the standard; in fact, the model provides an image of the analogue world (e.g., power system process, switchgear).

The IEC 61850 series defines the information and information exchange in a way that it is independent of a concrete implementation (i.e., it uses abstract models). The

standard also uses the concept of virtualisation (Figure 24). Virtualisation provides a view of those aspects of a real device that are of interest for the information exchange with other devices. Only those details that are required to provide interoperability of devices are defined in the IEC 61850.

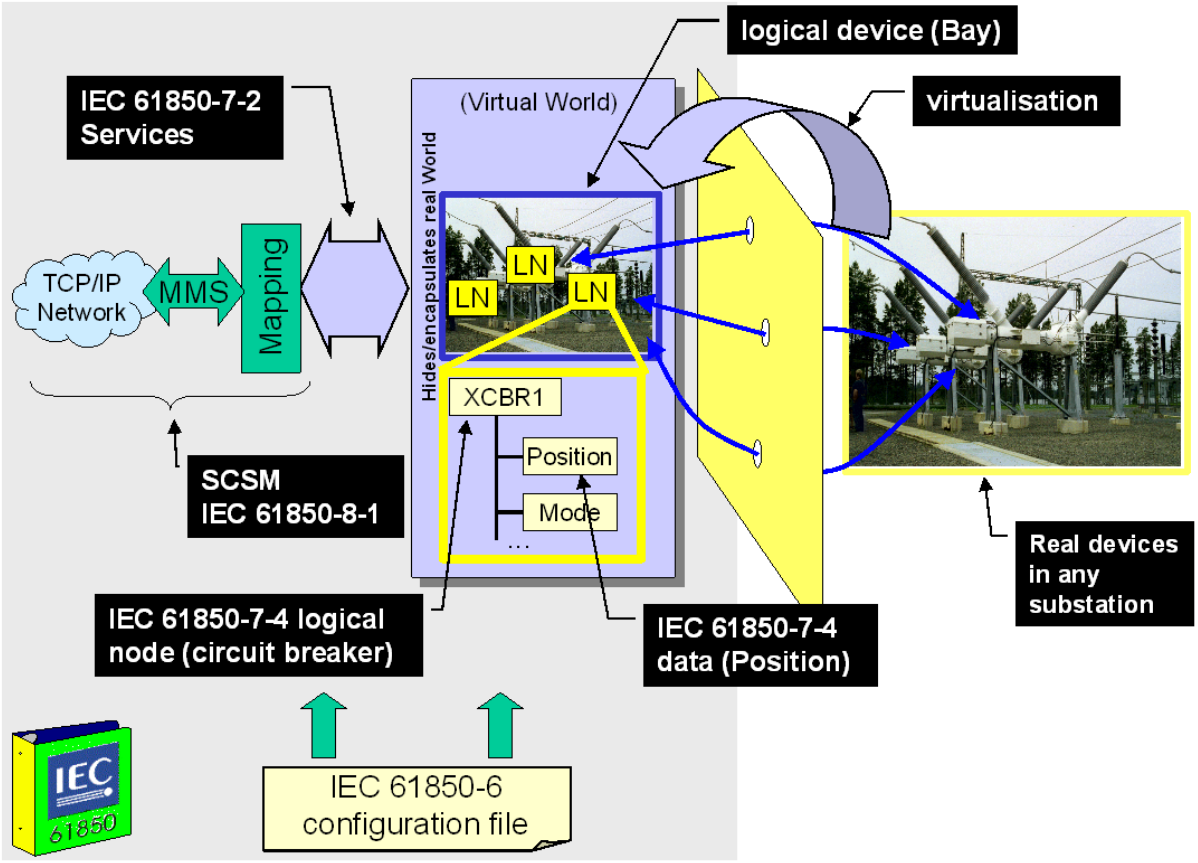


Figure 24 Virtualisation

The functions needed inside a substation are the base used to define the model of the data of IEC 61850. Each level of the substation has different functions with different priorities. Each function can be classified on control, monitor, protection, configuration or management task. Every function has some inputs, some outputs and several parameters[19][21][22][23][24].

IEC 61850 information model is based on two main levels of modelling – explained below (Figure 25):

- The breakdown of a real device (physical device) into logical devices

- The breakdown of logical device into logical nodes, data objects and attributes

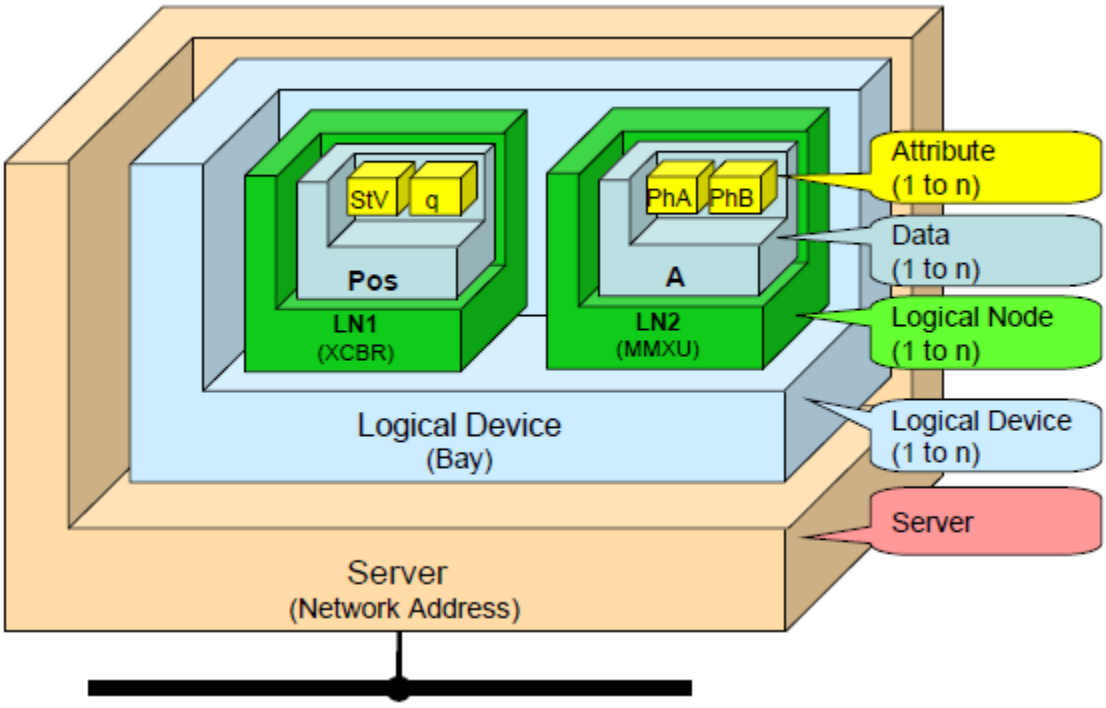


Figure 25 IEC 61850 modelling approach

Logical Device is the first level of breaking down the functions supported by a physical device (i.e. an IED). No specific rule is given by the standard on how to arrange Logical Devices into a physical device, except that one logical device can't be spread over many IEDs. It must be hosted by a single IED. Logical device usually represents a group of typical automation, protection or other functions.

The Logical Device hosts communication access point of IEDs and related communication services. It may have its own working mode and behaviour independent of other Logical Devices in a physical device. Logical devices provide information about the physical devices they use as host (nameplate and health) or about external devices that are controlled by the logical device (external equipment nameplate and health). Logical Device modelling concept helps modelling multifunction IEDs, gateway type IEDs or modular IEDs. It also enables the specification of a power utility automation system without having to specify any given product solution with physical devices.

The approach of the standard is to decompose the application functions into the smallest entities which are used to exchange information. The granularity is given by a reasonable distributed allocation of these entities to dedicated devices (IED). These entities are called Logical Nodes, each of which is identified with a four letter string (e.g., XCBR for a circuit breaker, PDIS for distance protection function, or MMXU for a measurement value).

Then several Logical Nodes build a Logical Device as defined above (e.g., a representation of a Bay unit). Logical Nodes included in a logical device may have a working mode that is different from that of the Logical Device they belong to. For example, an individual LN may have behaviour test/blocked without the entire Logical Device being so.

Based on its functionality, a Logical Node contains a list of Data (e.g. a position) with dedicated data attributes (further details will be provided in the following chapter). The data have a structure and a well-defined semantic (meaning in the context of systems for power utility automation or, e.g. more specifically, of substation automation systems) and are fully defined through IEC 61850-7.

The allocation of functions to devices (IEDs) and control levels is not fixed. The allocation normally depends on availability requirements, performance requirements, cost constraints, state of the art of technology, utilities' philosophies etc. Therefore, the standard should support any allocation of functions.

In order to allow a free allocation of functions to IEDs, interoperability is provided between functions to be performed in a power utility automation system but residing in equipment (physical devices in substation) from different suppliers. The functions may be split in modules performed in different IEDs but communicating with each other (distributed function). Therefore, the communication's behaviour of such a LN must support the requested interoperability of the IEDs.

A function is termed distributed when it is performed by two or more logical nodes that are in different physical devices (Figure 26). Since all functions communicate in some

way, the definition of a local or a distributed function is not unambiguous but depends on the definition of the functional steps to be performed until the function is completed.

When a distributed function is implemented, proper reactions on the loss of a functional component or an included communication link must be provided, e.g. the function may be blocked completely or shows a graceful degradation if applicable.

Figure 26 shows an example of two distributed functions, called respectively F1 and F2.

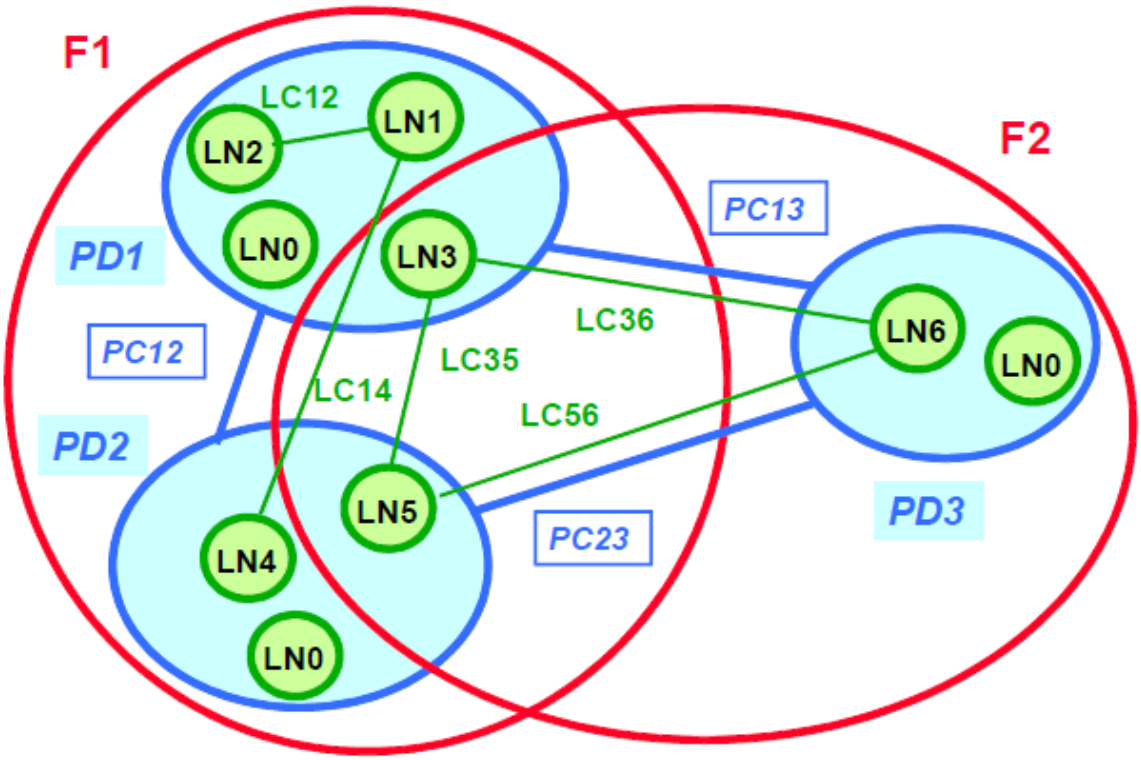


Figure 26 Example of IEC 61850 Distributed Functions

2.4.3 Logical Nodes, Common Data Classes and Attributes

Over one hundred logical nodes covering the most common applications of substation and feeder equipment are defined. While the definition of information models for protection and related applications is important because of the high impact of protection for safe and reliable operation of the power system, the covered applications include many other functions like monitoring, measurement, control and power quality.

There are logical nodes for automatic control, the names of which all begin with the letter A. There are logical nodes for metering and measurement the names of which all begin with the letter M. Likewise there are logical nodes for Supervisory Control (C), Generic Functions (G), Interfacing/Archiving (I), System logical nodes (L), Protection (P), Protection Related (R), Sensors (S), Instrument Transformers (T), Switchgear (X), Power Transformers (Y), and Other Equipment (Z).

Each logical node has an LN-Instance-ID as a suffix to the logical node name. For instance, suppose there were two measurement inputs in a device to measure two 3-phase feeders. The standard name of the logical node for a Measurement Unit for 3-phase power is MMXU. To delineate between the measurements for these 2 feeders the IEC 61850 logical node names of MMXU1 and MMXU2 would be used.

All the groups of logical nodes defined in IEC 61850 are showed in the following table.

Group indication	Description of the belonging group
A	Automatic control
C	Supervisory control
D	DER (Distributed Energy Resources)
F	Functional Blocks
G	Generic function references
H	Hydro power
I	Interfacing and archiving
K	Mechanical and non-electrical primary equipment
L	System logical nodes
M	Metering and measurement
P	Protection functions
Q	Power quality events detection related
R	Protection related functions
S	Supervision and monitoring
T	Instrument transformer and sensors
W	Wind power
X	Switchgear
Y	Power transformer and related functions
Z	Further (power system) equipment

Figure 27 LN groups defined by IEC 61850

Each logical node contains one or more elements of Data. Each element of data has a unique name. These Data Names are determined by the standard and are functionally related to the power system purpose. For instance, a circuit breaker is modelled as an XCBR logical node. It contains a variety of Data including Loc for determining whether operation is remote or local, OpCnt for an operations count, Pos for the position, BlkOpn block breaker open commands, BlkCls block breaker close commands, and CBOpCap for the circuit breaker operating capability.

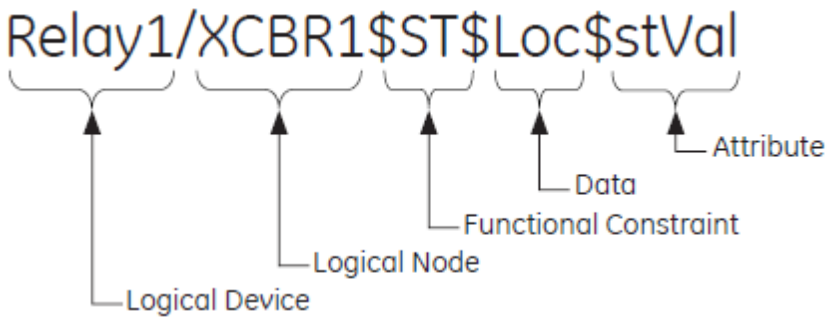


Figure 28 Example of IEC 61850 data modelling approach

The semantic of a logical node is represented by data and data attributes. Logical nodes may provide a few or up to 30 data. Data may contain a few or even more than 20 data attributes. Logical nodes may contain more than 100 individual information (points) organized in a hierarchical structure.

Most logical nodes provide data and data attributes that can be categorized in five categories:

- Common logical node information: information independent from the dedicated function represented by the LN (e.g., mode, health, name plate, etc.)
- Status information: representing either the status of the process or of the function allocated to the LN (e.g., switch type, switch operating capability, etc.)
- Settings: information needed for the function of a logical node (e.g., first, second, and third reclose time, close pulse time, and reclaim time of an auto-reclosing function)

- Measured values: are analogue data measured from the process or calculated in the functions like currents, voltages and power (e.g., total active power, total reactive power, frequency, net real energy since last reset, etc.)
- Controls: are data which are changed by commands like switchgear state (ON/OFF), tap changer position or resettable counters (e.g., position, block opening, etc.).

Each element of data within the logical node conforms to the specification of a Common Data Class (CDC) per IEC 61850-7-3. Each CDC describes the type and structure of the data within the logical node. IEC 61850-7-3 defines common data classes for a wide range of well-known applications. The core common data classes are classified into the following groups:

- Status information
- Measurand information
- Controllable status information
- Controllable analogue information
- Status settings
- Analogue settings
- Description information

Each CDC has a defined name and a set of CDC attributes each with a defined name, defined type, and specific purpose. Each individual attribute of a CDC belongs to a set of Functional Constraints (FC) that groups the attributes into categories.

For instance, in the Single Point Status (SPS) CDC described in Figure 29, there are functional constraints for status (ST) attributes, substituted value (SV) attributes, description (DC) attributes, and extended definition (EX) attributes. In this example the status attributes of the SPS class consists of a status value (stVal), a quality flag (q), and a time stamp (t).

IEDs are built up by composing logical nodes as depicted in Figure 30. The logical nodes are the building blocks of substation IEDs (e.g., circuit breaker XCBR and others). In the example for each phase, one instance of XCBR is used. The protection

IED receives the values for the voltage and current from conventional VT and CT. The protection functions in the protection device may detect a fault and issue or send a trip signal via the station bus. The standard supports also IEDs for digitizing VTs and CTs sending voltage and current as samples to the protection over a serial link. The output of conventional VTs and CTs may also be converted at the source to samples and transmitted over this serial link.

SPS Class					
ATTRIBUTE NAME	**ATTRIBUTE TYPE**	**FUNCTIONAL CONSTRAINT**	**TRGOP**	**VALUE / VALUE RANGE**	**MANDATORY/ OPTIONAL**
DataName	Inherited from Data Class (see IEC 61850-7-2)				
DATA ATTRIBUTE					
Status					
stVal	BOOLEN	ST	dchg	TRUE|FALSE	Mandatory
q	Quality	ST	qchg		Mandatory
t	TimeStamp	ST			Mandatory
Substitution					
subEna	BOOLEN	SV			PICS_SUBST
subVal	BOOLEN	SV		TRUE|FALSE	PICS_SUBST
subQ	Quality	SV			PICS_SUBST
subID	VISIBLE STRING64	SV			PICS_SUBST
Configuration, description and extension					
d	VISIBLE STRING255	DC		Text	Optional
dU	UNICODE STRING255	DC			Optional
cdcNs	VISIBLE STRING255	EX			AC_DLNDA_M
cdcName	VISIBLE STRING255	EX			AC_DLNDA_M
dataNs	VISIBLE STRING255	EX			AC_DLN_M

Figure 29 Single Point Status CDC

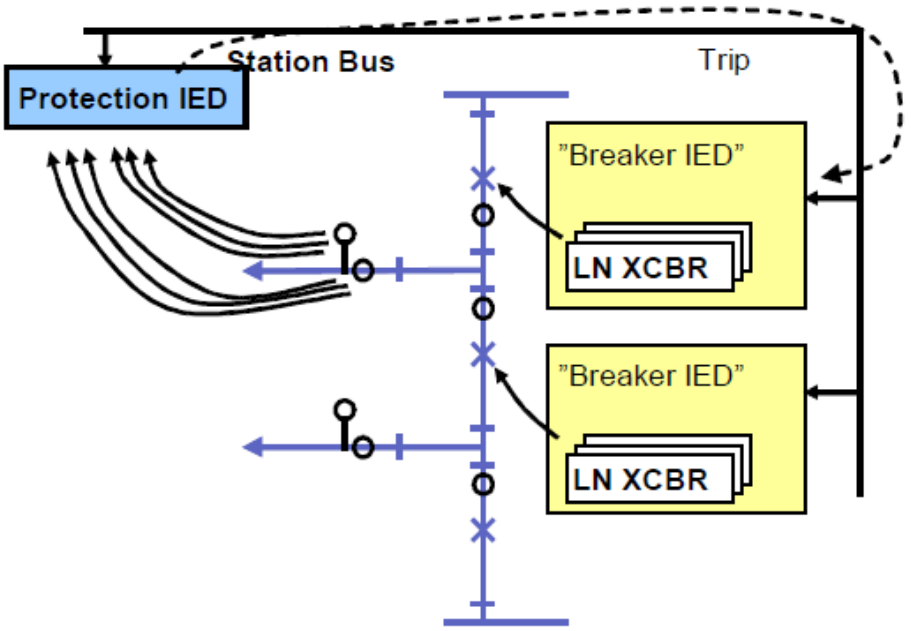


Figure 30 Build-up of devices (principle)

2.4.4 SCL

Engineering of a system normally starts before the system is physically available. In addition, modern IEDs are adaptable to many different tasks. However, this may not mean that all possible tasks can run in parallel at the same time, which leads to the situation that several capability subsets for the same device must be defined, each allowing to instantiate/use all the contained capabilities.

Therefore, although the devices may be self-descriptive, the device capabilities as well as their project specific configuration in general and with respect to the system parameters should be available in a standard way before the IED itself is available and engineered. To be able to exchange the device descriptions and system parameters between tools of different manufacturers in a compatible way, IEC 61850-6 defines a System Configuration description Language (SCL). This language allows:

- System functional specification
- IED capability description
- Power Utility automation system description

These functions provide standardized support to system design, communication engineering and to the description of readily engineered system communication for device engineering tools, during the whole life cycle of the installation. The SCL language itself is based on eXtensible Markup Language (XML).

Below is a list of the different types of SCL files specified by the standard:

- ICD for IED Capability Description
- SSD for System Specification Description
- SCD for Substation Configuration Description
- CID for Configured IED Description

The default functionality of an IED in the distribution substation configuration language is represented by the IED Capability Description (ICD) file. It is used for data exchange from the IED configuration tool to the system configuration tool. This ICD file describes the capabilities of an IED. It contains exactly one IED section for the IED whose capabilities are described. The file also includes the different logical node types as they are instantiated in the device. The file extension is .icd for IED Capability Description.

The description of the system is the first step in the engineering process and until now has not been based on any standardized approach yet. The IEC 61850 engineering process envisions the use of distribution substation specification tools that allow the user to describe the distribution substation design and associated functional requirements for the protection and automation systems. The data exchange from such a system specification tool and other tools utilized in the process should be based on the System Specification Description files. The SSD file describes the single line diagram of the substation and the functional requirements represented by logical nodes.

The configuration of the system is represented by the distribution Substation Configuration Description (SCD) file. It contains distribution substation description section, communication configuration section and all IEDs. The IEDs in the SCD file are not anymore in their default configuration, but as they are configured to operate

within the distribution substation protection and automation system. These files are then used to configure the individual IEDs in the system.

The difference between the IED Capability Description (ICD) file and the Configured IED Description (CID) file is that the second includes the distribution substation specific names and addresses instead of the default ones in the first. The CID file represents a single IED section of the SCD file described above.

Figure 31 shows a possible use of the SCL files explained above.

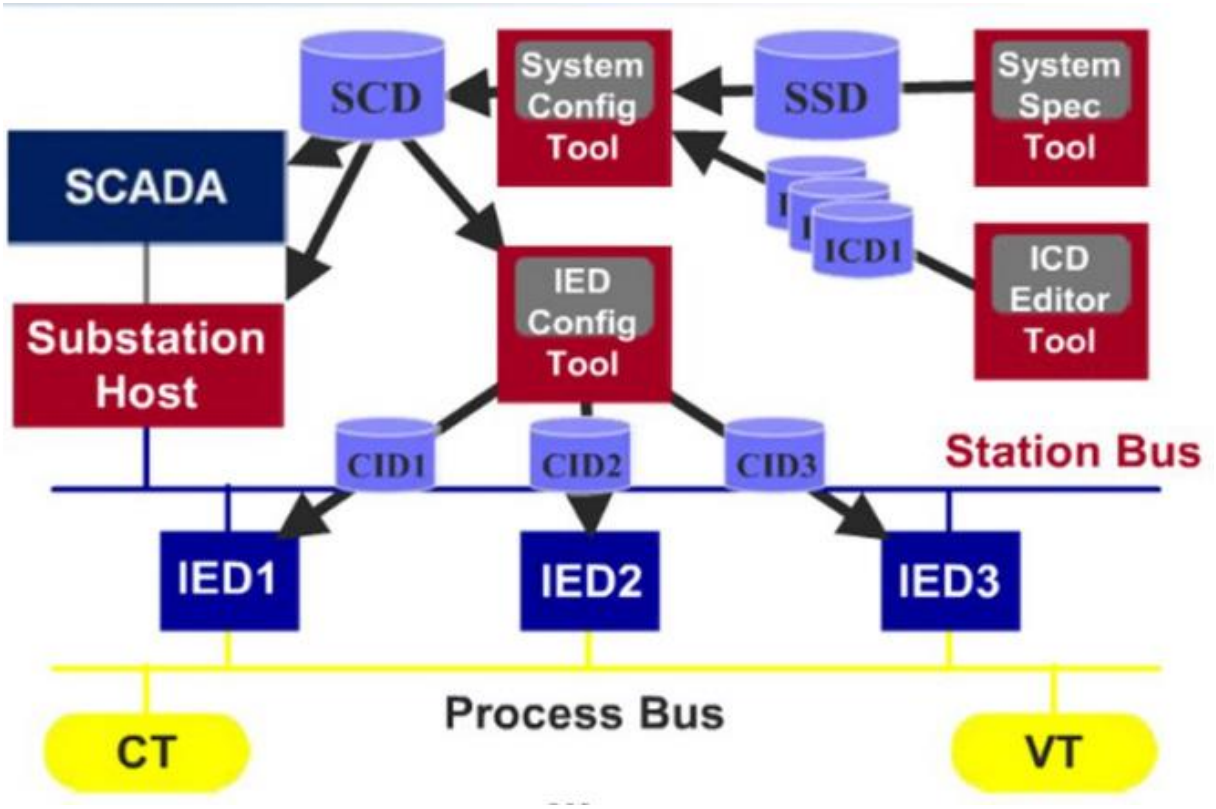


Figure 31 Possible use of SCL files

2.4.5 IEC 61850 communication protocols

The OSI Reference Model (ISO/IEC 7498-1) details a model based upon the concept of layering of communication functionality. The model defines 7 layers and details the functional requirements for each layer, in order to achieve a robust communication system. The model does not specify the protocols to be used to achieve such a functionality, nor it restricts the solution to a single set of protocols.

The use of ISO Application (A-Profile) and Transport (T-Profile) Profiles (Figure 32) describes the various stack profiles. An ISO A-Profile is the set of specifications and agreements relating to the upper 3 layers of the ISO OSI reference model (i.e. the layers of application, presentation, and session). An ISO T-Profile is the set of specifications and agreements relating to the lower 4 layers of the ISO OSI reference model (i.e., the layers of transport, network, DataLink and physical).

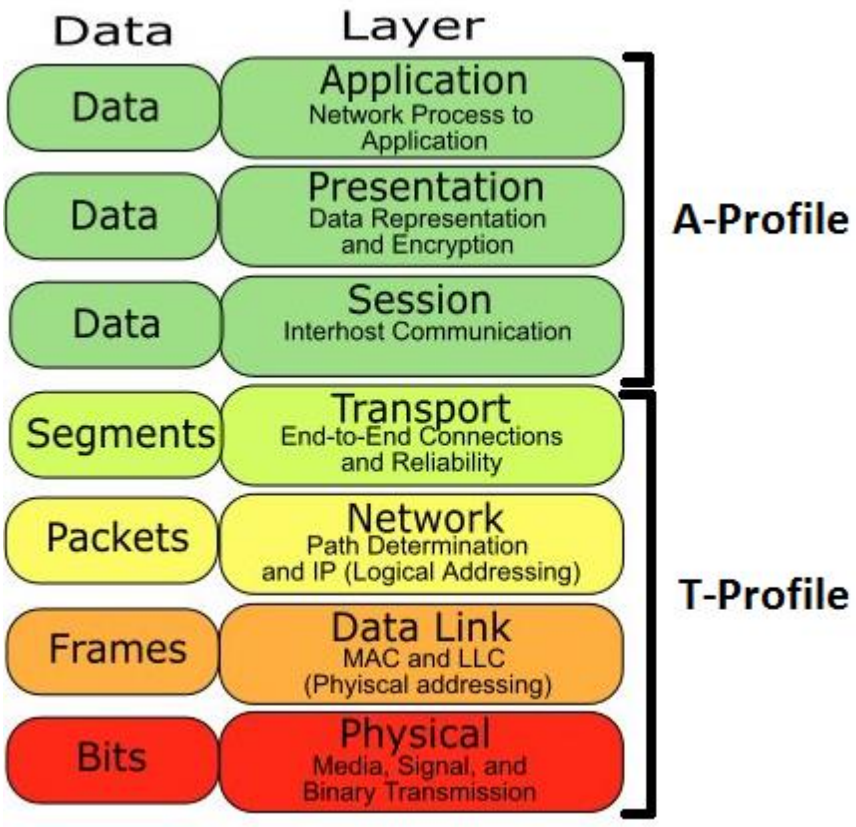


Figure 32 ISO/OSI model

The abstract data and object models of IEC 61850 define a standardized method of describing power system devices that enables all IEDs to present data using identical structures that are directly related to their power system function.

The Abstract Communication Service Interface (ACSI) models of IEC 61850 define a set of services and the responses to those services that enables all IEDs to behave in an identical manner from the network point of view. While the abstract model is critical to achieve this level of interoperability, these models need to be operated over a real

set of protocols that are practical to implement and that can operate within the computing environments commonly found in the power industry.

Each communication defined in IEC 61850 uses one of the two following communication mechanisms:

- Client-Server: This model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, i.e. servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.
- Publisher-Subscriber: In this messaging pattern, senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers. Instead, the publishers characterize published messages into classes without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are.

Referring to Figure 33, the message types and performance classes specified in IEC 61850 are listed below:

- Type 1 (Fast messages)
- Type 1A (Trip)
- Type 2 (Medium speed messages)
- Type 3 (Low speed messages)
- Type 4 (Raw data messages)
- Type 5 (File transfer functions)
- Type 6 (Time synchronization messages)

Messages of Type 1 and Type 1A are mapped directly into the Ethernet layer, while messages of Type 2, 3, and 5 require message oriented services, such that the Manufacturing Message Specification (MMS) standard is used.

The IEC 61850 standard enables information exchange through different communication services (Figure 33):

- Sampled Values (SV) are encapsulated and transmitted as a multicast service over the Ethernet providing fast and cyclic exchange of voltage and current measurement values for protection and control replacing traditional analog wiring. Any sample loss or a delay longer than 4 ms between two consecutive samples prevents IEDs from functioning correctly. For example, the bus-bar voltage used to trigger protection relays is measured at 4000 samples/s and transmitted cyclically at 1 kHz. MAC-layer multicast addressing is used to make the sample values available to multiple users. These messages use publisher-subscriber mechanism (i.e., horizontal communication).
- GOOSE (Generic Object Oriented Substation Event) is used for transmission of critical events in real time (e.g., tripping commands) between two or more IEDs using the Ethernet multicast. The GSE (Generic Substation Event) service model of IEC 61850-7-2 details fast and reliable system-wide distribution of input and output data values using a specific scheme of retransmission to achieve the appropriate level of reliability (discussed in the following sections). These messages use publisher-subscriber mechanism.
- TCP/IP-based messages using MMS (Manufacturing Messaging Service) data presentation layer are employed for download and upload of configuration, parameter setting, monitoring, etc. These messages use client-server mechanism (i.e. vertical communication).

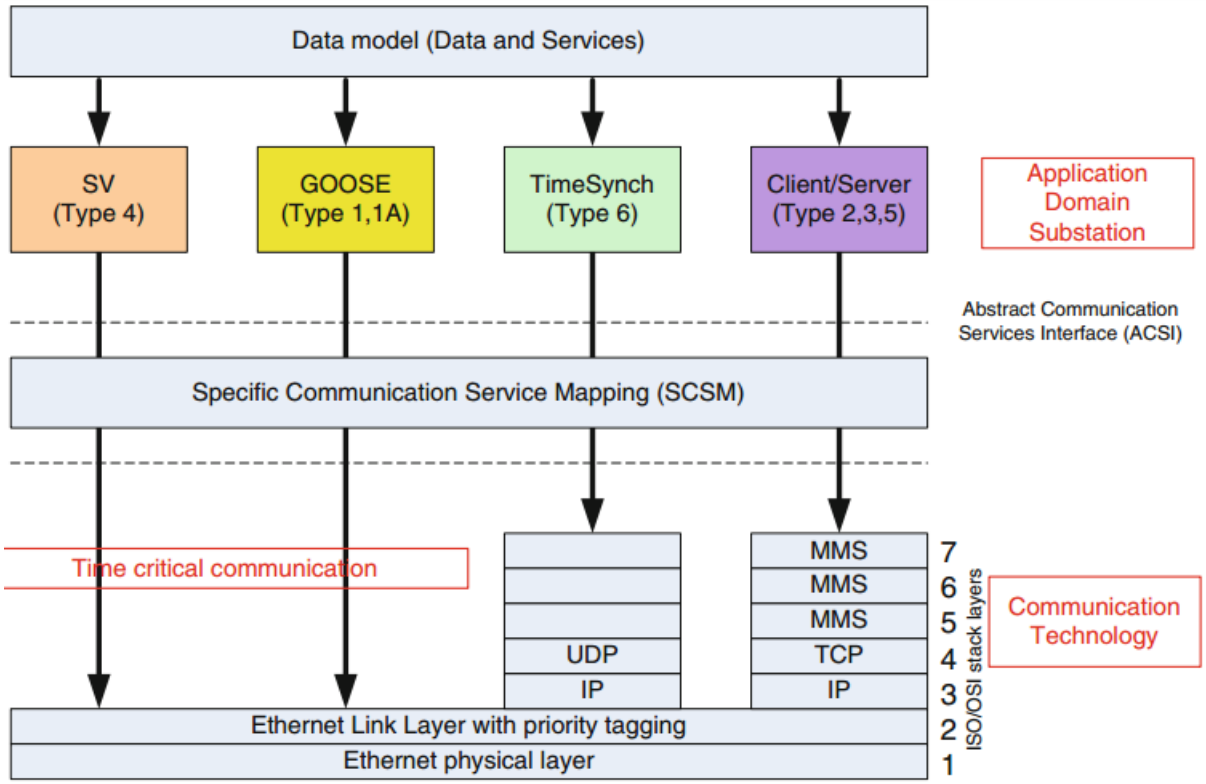


Figure 33 Overview of IEC 61850 Functionality and Associated Communication Profiles

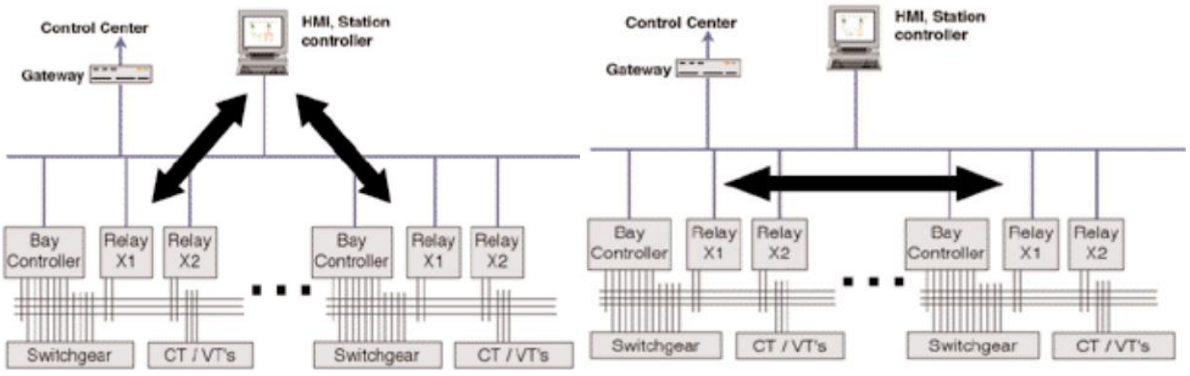


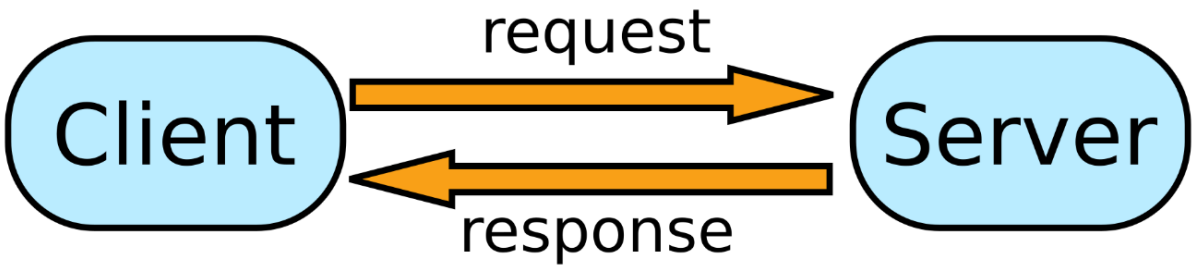
Figure 34 Vertical communication on the left and horizontal communication on the right

Control blocks defined for reporting, logging, GSE and SV are modelled like data object classes. There is however a fundamental difference between data objects and control blocks.

Data objects are used to interface with application level functions, while above-mentioned control blocks configure the communication services. The configuration language (SCL) separates the data objects for the application information completely

from the communication service models (defined as control blocks). In this regard, the GOOSE control block will be analysed later.

2.4.6 Client-Server based IEC 61850 communication



Supervisory control and data acquisition (SCADA) are one of the basic tasks of a substation automation system. This comprises:

- Local and remote operation of the switchgear and other high voltage equipment
- Acquisition of switchgear information and power system measurands
- Handling of events and alarms.

The SCADA application is related to human operation of the network and is performed by a local or remote operator. The data communication for this application is directed vertically, i.e. from a higher hierarchical control level down to a lower one (commands of any kind from the operator's place) or reverse (binary indications like breakers or isolators position, measurands from instrument transformers and other sensors, events, alarms). It allows to operate and to supervise the power system.

For this vertical relationship, IEC 61850 is using the client - server concept.

The server is the process or bay level IED, which provides all data to the client at station or any remote level. The data are provided on request by the server or automatically by a report from the server issued if certain conditions are fulfilled.

Reporting and logging as well as the basic services of the data model provide flexible data retrieval schemas, for example:

- Change-of-state notification of clients: immediate reports

- Sequence-of-events: keeping reports in sequence or storing and querying sequences of log entries
- Polling data objects at any time: GetDataValues and GetDataSetValues.

In a client-server communication, the client controls the data exchange. Therefore, client-server communication is very flexible in terms of the data to be transmitted. Compared to a master-slave system, the client-server concept allows the implementation of multiple clients in the same system.

The data-set (referencing data object) represents the values of data objects. The values are conceptually monitored by the event monitors. An event monitor determines, based on the state of the real data and the attributes of the control class, when to generate a notification to the appropriate handlers (for example log or report handler). This notification includes the data object values and reasons for data inclusion.

The report-control-block manages the procedures that are required for reporting values of data objects from one or more logical node to one client. There are two classes of report control blocks defined, each with a slightly different behaviour:

- Buffered-report-control-block (BRCB): internal events (caused by trigger options data-change, quality-change, and data-update) issue immediate sending of reports or buffer the events (to some practical limit) for transmission, such that values of data object are not lost due to transport flow control constraints or loss of connection. BRCB provides the sequence-of-events (SOE) functionality;
- Unbuffered-report-control-block (URCB) – internal events (caused by trigger options data-change, quality-change, and data-update) issue immediate sending of reports on a “best efforts” basis. If no association exists, or if the transport data flow is not fast enough to support it, events may be lost.

Figure 35 shows a quick overview of the most important attributes for a generic RCB (Report Control Block).

ID	Full Name	Description
RptID	Report ID	It is the client-specified report identifier of the RCB that has caused the generation of the report. This field may be used by clients to distinguish between reports from various RCBs.
RptEna	Report Enabled	It is used by a client to enable and disable reporting, and it indicates this state in the RCB.
DatSet	Data Set	It specifies the ObjectReference of the data-set being monitored and whose values of the members of the data-set (one, a subset, or all) are reported.
TimeOfEntry	Time Stamp	This attribute represent a Time Stamp indicating the time at which the internal event notification was received by the report handler.
TrgOps	Trigger Options	The attribute TrgOps specifies the trigger conditions which are monitored by that specific RCB. Below is a list of the available conditions: • Data-Change • Quality-Change • Data-Update • Integrity • General Interrogation
SqNum	Sequence Number	The attribute SqNum specifies the sequence number for each RCB that has report enable set to TRUE. This number is to be incremented by the RCB for each report generated and sent.

Figure 35 Example of some attributes available for RCBs

Client-server communication relies on the full seven-layer-stack using a confirmed transmission layer and is, as consequence, very reliable but relatively time consuming. Therefore, the client-server communication is not suited for time-critical data transmission but very well for the communication with an operator having a response time of the order of 1 s.

An important aspect of a data model is the unambiguous identification of the data. According to IEC 61850, the name is created by the concatenation of the individual elements of the hierarchical data model: logical device, logical node instance, data and data attribute. For example, the status of an IED (a circuit breaker in this case) is provided by Figure 36.

To access the data, several services are standardized as part of the client-server concept. Besides the basic services to access the data model (individual read or write of data), more elaborated services are defined.

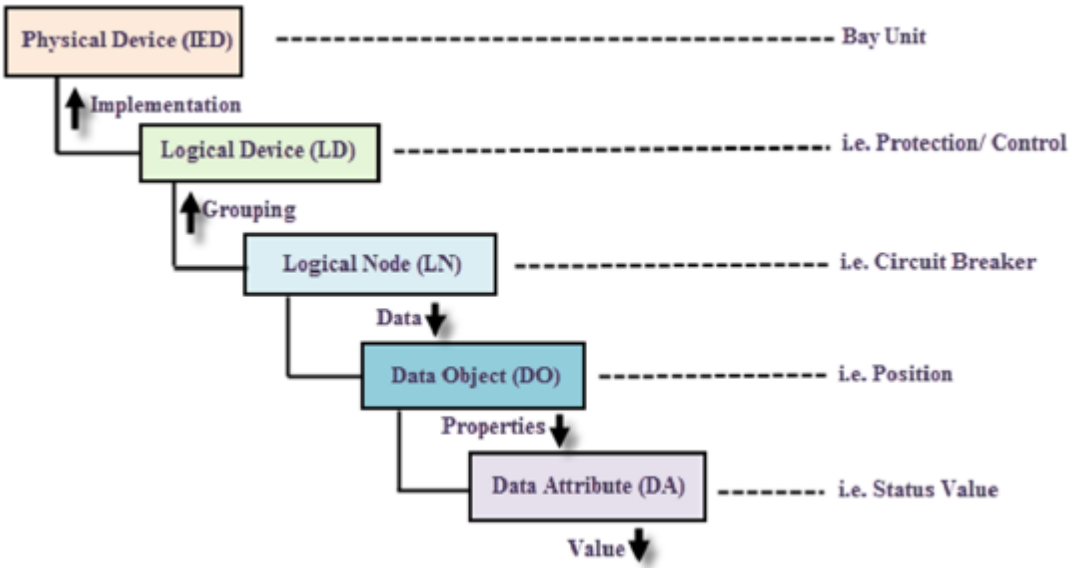


Figure 36 Hierarchical data model and naming

As an example, for the SCADA application, event driven transmission of data is essential. In IEC 61850, the report service is defined for that purpose. The report service is not accessing individual data, but group of data called dataset. The details of the event driven transmission are defined in the report configuration block. An event causing a transmission may be a change of a binary value, the crossing of a predefined alarm limit or the expiration of a cycle time.

Based on the report configuration block and the related dataset, reports are sent to the client. Including the time tag of the event as part of the dataset, event lists can be created. For that purpose, the IED is typically synchronized with accuracy in the order of 1 ms.

2.4.7 Publisher-Subscriber based IEC 61850 Communications

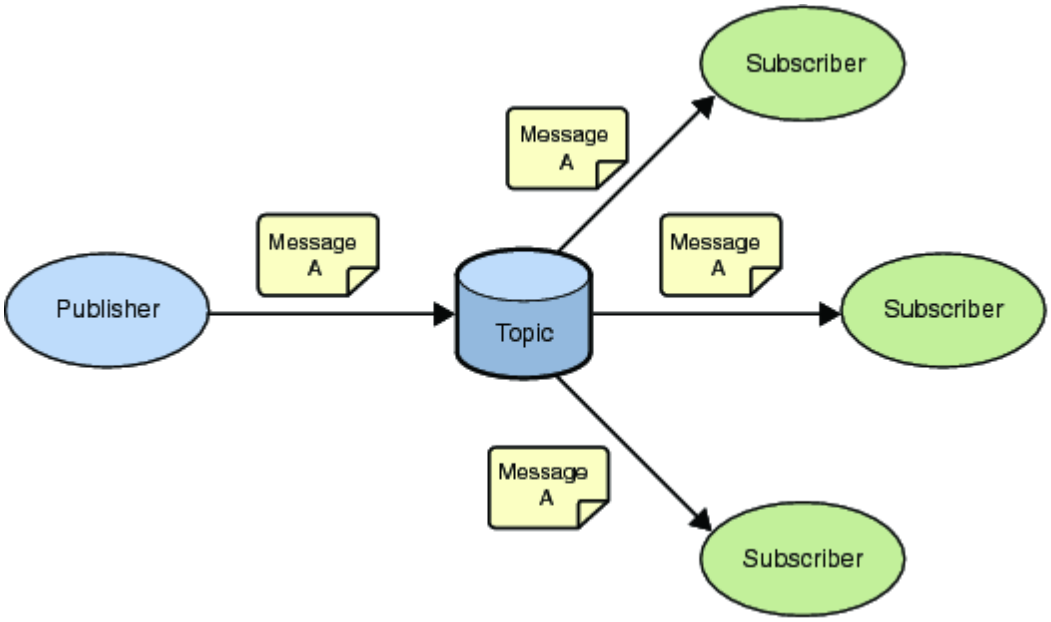


Figure 37 Publisher-Subscriber mechanism

There are several automated functions in the substation automation system, which require a time critical exchange of binary information between functions located within the same bay or in different bays. Examples:

- Exchange between line protection and auto-recloser
- Exchange between bays for breaker failure
- Exchange between bays for station interlocking

Typically, operators don't interact with these functions. They are time critical because they are safety critical. The maximal accepted communication delay is in the range of several milliseconds.

If the functions exchanging the information are in different IED's, the information exchange may be done using copper wiring with contacts and auxiliary relays or using serial communication. This information exchange is a horizontal communication between devices at the same hierarchical level.

Theoretically, the information exchange could take place using the client-server communication. However, client-server communication is using the full seven-layer

stack and is therefore relatively time consuming. An appropriate communication concept used is a publisher-subscriber communication. The publisher is distributing the information over the communication network; the subscriber may receive the information according his needs.

In IEC 61850, publisher-subscriber communication is not using confirmed services and is therefore transmitted over a reduced communication stack resulting in a very short transmission time. GOOSE and SV messages belong to this category.

The generic substation event model is based on the concept of an autonomous decentralization, providing an efficient method allowing the simultaneous delivery of the same generic substation event information to more than one physical device using multicast/broadcast services.

As the information exchange is based on a publisher-subscriber mechanism, the publisher writes the values into a local buffer at the sending side and the subscriber reads those values from a local buffer at the receiving side. The communication system is responsible to update the local buffers of the subscribers. A generic substation event control class in the publisher is used to control the procedure.

If the value of one or several DataAttributes of a specific functional constraint (for example ST) in the dataSet changes, the transmission buffer of the publisher is updated and all values are transmitted with a GOOSE message. Mapping specific services of the communication network will update the content of the buffer in the subscribers. New values received in the reception buffer are signalled to the application. The GOOSE messages contain information that allow the receiving device to know that a status has changed and the time of the last status change. The time of the last status change allows a receiving device to set local timers relating to a given event. A newly activated device, upon power-up or reinstatement to service, sends the current value of a data object (status) or values as the initial GOOSE message. Moreover, all devices sending GOOSE messages continues to send the message with a long cycle time, even if no status/value change has occurred. This ensures that devices that have been activated recently will know the current status values of their peer devices.

The GOOSE communication uses a specific scheme of re-transmission to achieve the appropriate level of reliability. When a GOOSE server generates a SendGOOSEMessage request, the current data set values are encoded in a GOOSE message and transmitted on the multicast association. Additional reliability is achieved by re-transmitting the same data with an exponential scheme. How this will be done is shown in Figure 38, where:

- T_0 (also known as $T_{\max}$) is the retransmission time in stable conditions (no event for a long time)
- (T_0) is the retransmission time in stable conditions when shortened by an event
- T_1 is the shortest retransmission time after the event (also known as $T_{\min}$)
- T_2 and T_3 are the retransmission times until achieving the stable conditions time

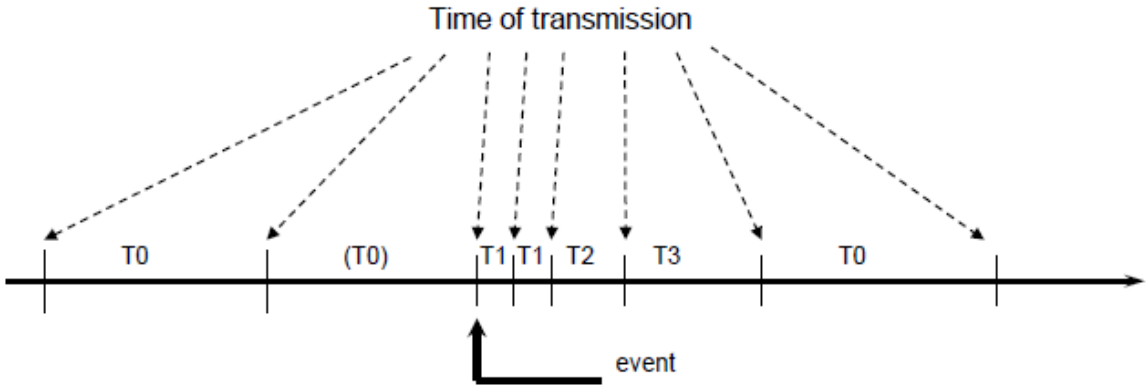


Figure 38 GOOSE re-transmission scheme

Below is a quick overview of the GOOSE Control Block (GoCB) for each specific GOOSE association shown in Figure 39, a detailed view of such parameters is provided later.

- GoCBName: It unambiguously identify a GoCB within the scope of a LLN0
- GoCBRef: It is the unique path-name of a GoCB within the LLN0
- GoEna: It indicates whether that GoCB is currently enabled to send GOOSE messages or not
- GoID: it is a user definable identification of the GOOSE message

- DataSet: it represents the reference of the data-set whose values of members are transmitted
- ConfRev: it represents a count of the number of times that the configuration dataSet has been changed
- NdsCom: Its value is TRUE if the GoCB requires further configuration
- DstAddres: It is the SCSM specific addressing information like media access address, priority, and other information.

Attribute name	Attribute type	Value/explanation
GoCBName	ObjectName	Instance name of an instance of GoCB
GoCBRef	ObjectReference	Path-name of an instance of GoCB
GoEna	Boolean	Enabled (TRUE) | disabled (FALSE)
GoID	Visible-string (ASCII)	Attribute that allows a user to assign an identification for the GOOSE message
DatSet	ObjectReference	
ConfRev	Integer	
NdsCom	Boolean	
DstAddress	Phisical address	

Figure 39 GoCB parameters

For the GOOSE communication, the ACSI defined the following list of services:

- SendGOOSEMessage: It is used by a GoCB to send a GOOSE message over multicast-application-association
- GetGoReference: The members of the data-set are uniquely numbered beginning with 1. This number is called the MemberOffset of a given member. Each member of the data-set has a unique number and a MemberReference (the Functional Constraint Data or Functional Constraint Data Attribute). This service retrieves the MemberReferences of specific members of the DATA-SET of the referenced GoCB
- GetGOOSEElementNumber: it retrieves the member position of a selected DataAttribute in the data-set associated with a GoCB.
- GetGoCBValues: It is used to retrieve attribute values of GoCB made visible and thus accessible to the requesting client by the referenced LLN0
- SetGoCBValues: It is used to set attribute values of GoCB made visible and thus accessible to the requesting client by the referenced LLN0.

2.4.8 IEC 61850 GOOSE Communication

As seen in the previous section, the GOOSE messages are encapsulated directly into an Ethernet frame. The analysis of the Ethernet structure can be found in section 2.2.3. Below is an analysis of all the Ethernet parameters that are typical for GOOSE communication[16]:

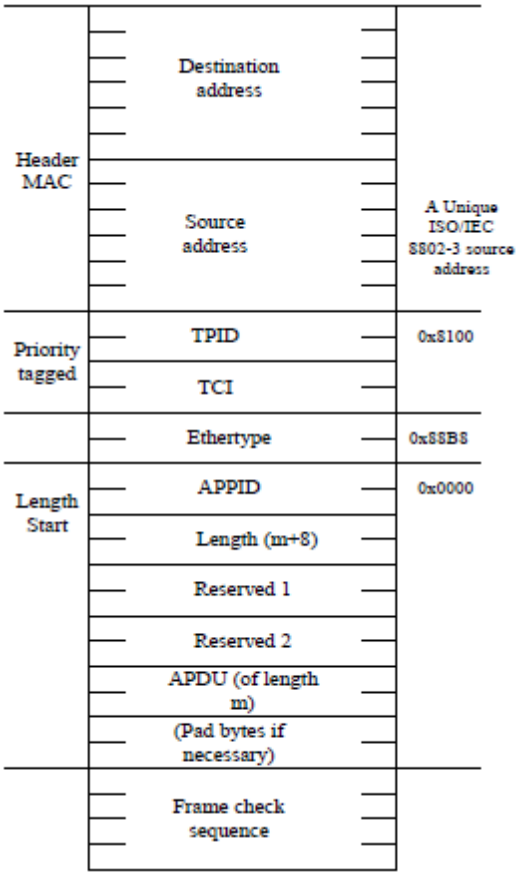


Figure 40 Structure of a GOOSE message

- Destination Address: as GOOSE message are usually sent through multicast, this field contains a multicast address. For this purpose, IEEE reserved the multicast addresses from 01-0C-CD-01-00-00 to 01-0C-CD-01-01-FF for GOOSE communication (Figure 41).

Ethernet multicast address	Usage
From 01-0C-CD-01-00-00 To 01-0C-CD-01-01-FF	IEC 61850-8-1 GOOSE Type 1/1A
From 01-0C-CD-02-00-00 - To 01-0C-CD-02-01-FF	GSSE (IEC 61850 8-1)
From 01-0C-CD-04-00-00 - To 01-0C-CD-04-01-FF	Multicast sampled values (IEC 61850 8-1)

Figure 41 Reserved multicast address for IEC 61850 purposes

- VLAN Tag: GOOSE messages are IEEE 802.1Q frames. Figure 42 shows default values for IEC 61850 frames.

Service	Default VID	Default priority
GOOSE	0	4
GSE	0	1
Sampled Values	0	4

Figure 42 IEC 61850 default VID and Priority

- Payload: In the GOOSE communication it contains:
    - Application Identifier (APPID): The APPID is used to select ISO/IEC 8802-3 frames containing GSE Management and GOOSE messages and to distinguish the application association. The value of APPID is the combination of the APPID Type, defined as the two most significant bits of the value (Figure 43), and the actual ID. The reserved value range for GOOSE Type 1 is 0x0000 to 0x3FFF, for GOOSE Type 1A (Trip) the reserved value range is 0x8000 to 0xBFFF. If no APPID is configured, the default value shall be 0x0000. The default value is reserved to indicate lack of configuration.

Use	Ethertype value (hexadecimal)	APPID type
IEC 61850-8-1 GOOSE Type 1	88-B8	0 0
IEC 61850-8-1 GSE Management	88-B9	0 0
IEC 61850-9-2 Sampled Values	88-BA	0 1
IEC 61850-8-1 GOOSE Type 1A	88-B8	1 0

Figure 43 Ethertype and APPID values for GSE communications

- Length: it indicates the number of octets including the EtherType PDU header starting at APPID, and the length of the APDU (Application Protocol Data Unit). Therefore, the value of Length is equal to 8 + m, where m is the length of the APDU and m is less than 1492.
- Reserved 1: The structure of this field is shown in and contains:
  - Simulated (S): When the bit S is set, the GOOSE telegram has been issued by a publisher located in a test device and otherwise its value is 0.
  - Reserved (R): The three bits are reserved for future application, hence they are equal to 0.
  - Reserved security: the Reserved security field is defined by the security standard IEC 62351-6 and is used as defined when GOOSE with security is transmitted, otherwise its value is 0.

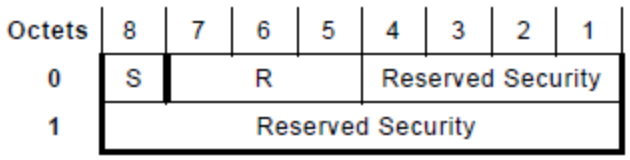


Figure 44 Reserved 1 fields

- Reserved 2: the Reserved 2 field is defined by the security standard IEC 62351-6 and is used as defined when GOOSE with security is transmitted otherwise its value is 0.
- APDU: It is the Application Protocol Data Unit and contains the actual information which is sent through the GOOSE message (its content will be analysed with a practical example in the following).

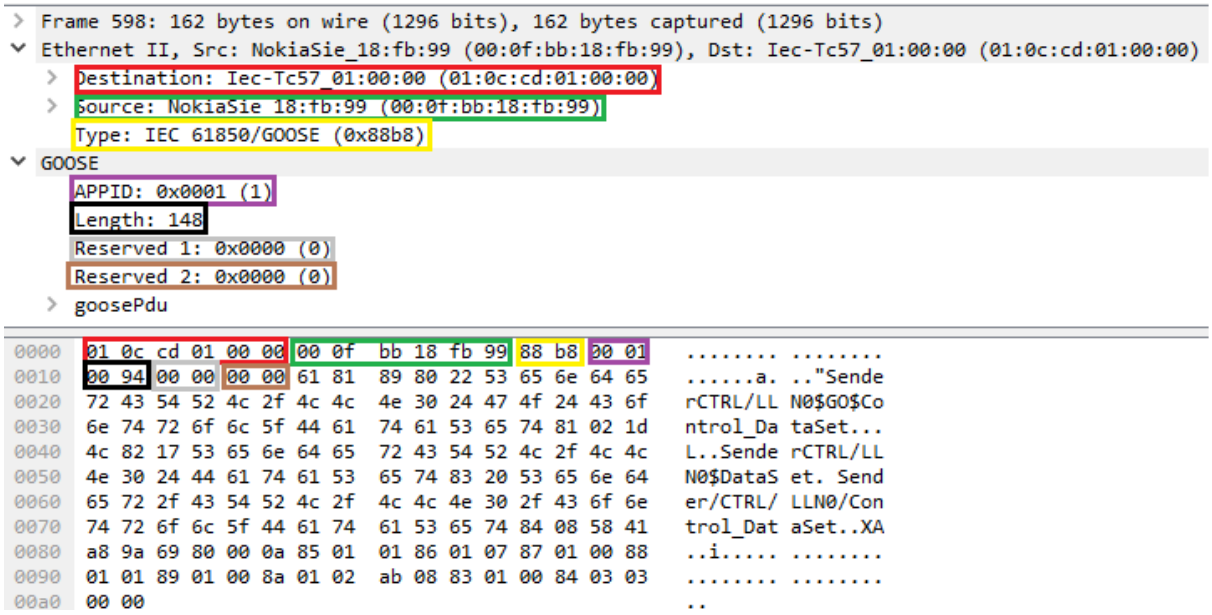


Figure 45 Captured GOOSE message (header)

In order to analyse the content of a GOOSE message (APDU), a real GOOSE was captured through Wireshark (in this example, Ethernet frame without IEE E802.1Q Tag was used).

Some of its parameter have been explained in the previous section, below is a full overview of the content of the fixed part of such messages:

- Destination MAC address: is one of the multicast MAC address reserved for GOOSE applications (red square in the captured GOOSE message)
- Source MAC address: represents MAC address of the IED which has generated the Ethernet packet (green square in the captured GOOSE message)
- EtherType: it represents the hex value for the EtherType as previously stated (yellow square in the captured GOOSE message)
- APPID: it represents the value associated to that specific application as previously stated (violet square in the captured GOOSE message)
- Length: it represents the length as previously stated (black square in Figure 45)
- Reserved 1: as no security features were used, its value is equal to 0 (grey square in the captured GOOSE message)
- Reserved 2: as no security features were used, its value is equal to 0 (grey square in the captured GOOSE message)

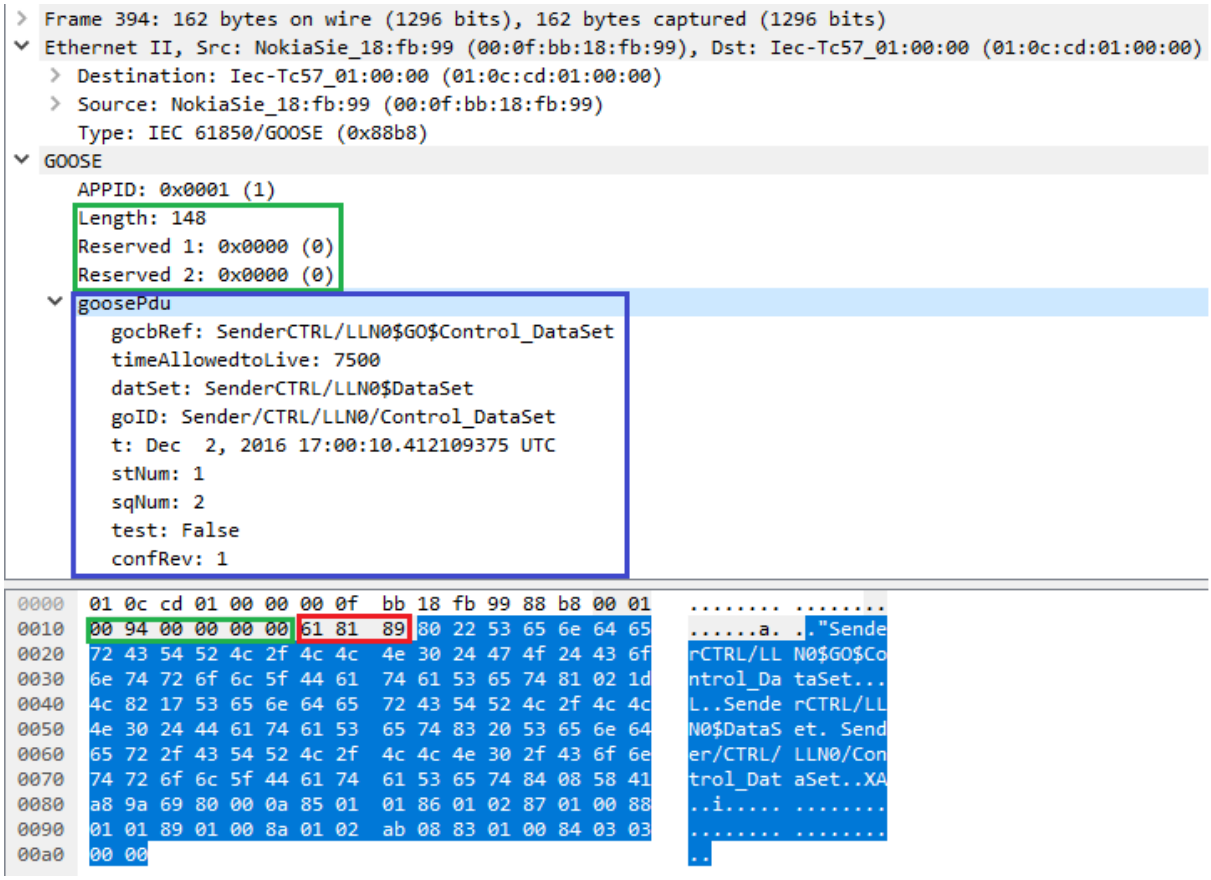


Figure 46 Captured GOOSE message overview

Figure 46 shows the first part (as seen before) and the second part (blue square). This second part follows ASN.1 Basic encoding rules (as specified in ISO/IEC 8825-1) to encode and decode the GOOSE telegram. The main encoding principle is below.

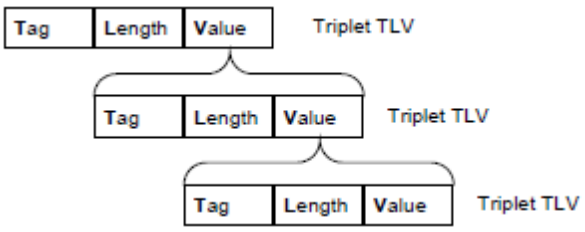


Figure 47 Basic Encoding Rules format

Therefore, each value content in this last part is composed respectively by the following three fields:

- Tag: is 1 byte long and represent the value corresponding to a specific GOOSE parameter, it is used to understand the content of the following bits. The match

between the hexadecimal value and the actual tag is defined within the standard.

- Length: is 1 byte long and represent the length of the data field for that specific parameter
- Data: is the actual value of the parameter, its length is defined in the previous field.

The example provided in Figure 46 constitutes the only exception to the stated rules. Particularly, in between two of the parts defined before, there is a 3-byte long string (red square) which does not follow the usual rule (an additional length is present):

- The first bit is equal to 61 and represents the tag (goosePdu)
- The second bit is an additional length compliant with the Basic Encoding Rules (BER)
- The third represent the length in byte of the goosePdu (in this example, the hex value is equal to 89, i.e. 137 in the decimal system)
- The payload containing the data value (blue square)

Below is a detailed view of all the possible parameters contained in the variable part such messages:

- GOOSE Control Block Ref: the attribute GoCBRef is the unique path-name of a GoCB within the LLN0. The GoCBRef follow the rule: LDName/LLN0.GoCBName. This visible-string has a maximum size of 129 octets. The value is the reference to the associated GoCB that is controlling the GOOSE message. This field must be read using the ASCII standard.
- Time Allowed to Live: each message in the retransmission sequence carries a TAL parameter that informs the receiver of the maximum time to wait for the next re-transmission. If a new message is not received within that time interval, the receiver assumes that the association is lost.
- datSet: the attribute DatSet represents the reference of the data-set whose values of members shall be transmitted. The members of the data-set are uniquely numbered beginning with 1. This visible-string has a maximum size of

129 octets and its value is the same as found in the associated GoCB specified by DatSet. This field must be read using the ASCII standard.

- GOOSE ID: the attribute GoID is a user definable identification of the GOOSE message. This visible-string has a maximum size of 65 octets. The value is the same as found in the associated GoCB specified by GoID. This field must be read using the ASCII standard.
- Time Stamp: the parameter T contains the time at which the attribute StNum was incremented. This field uses the UTC time type through an 8-byte long string and encoded as defined in RFC 1305. The integer part contains: elapsed number of whole seconds since GMT midnight January 1, 1970(s). The fractional part contains the portion of a second elapsed since the last whole second.
- State Number: the parameter StNum contains the counter that increments each time a GOOSE message has been sent and a value change has been detected within the data-set specified by DatSet. This integer value has a range of 1 to 4 294 967 295.
- Sequence Number: the parameter SqNum contains the counter that increments each time a GOOSE message has been sent. Following a StNum change, the counter SqNum is set to a value of 0. Integer value a range of 0 to 4 294 967 295, where 0 is reserved and used to indicate the first transmission due to a change in StNum.
- Test: the parameter simulation indicates with the value TRUE that the message and therefore its value have been issued by a simulation unit. The GOOSE subscriber will report the value of the simulated message to its application instead of the “real” message depending on the setting of the receiving IED.
- ContRev: this Boolean value indicates if a configuration change has occurred. A value of TRUE indicates that a configuration change has been detected. Upon detection, the value remains TRUE for a certain amount of time (from 30 sec to 60sec, decided by the manufacturer)
- NdsCom: The attribute NdsCom has a value of TRUE if the GoCB requires further configuration.

- allData: This field contains the user defined information following the BER rules explained above.

Figure 49 provides a detailed view of the parameters listed above in a real example, where all the fields are summarized in Figure 48[11].

Name	B.E.R. TAG	Length (Dec)	Encoding of data field	Reference colour
gocbRef	80	22 (34 byte)	Visible-string (ASCII)	Blue
TAL	81	02 (2 byte)	INT32U	Red
dataSet	82	17 (23 byte)	Visible-string (ASCII)	Violet
goID	83	20 (32 byte)	Visible-string (ASCII)	Yellow
Time Stamp	84	08 (8 byte)	UTC Time	Black
State Number	85	01 (1 byte)	INT32U	Orange
Sequence Number	86	01 (1 byte)	INT32U	Brown
test	87	01 (1 byte)	Boolean	Pink
contRev	88	01 (1 byte)	INT32U	Light green
ndsCom	89	01 (1 byte)	Boolean	Grey
numDataSetEntries	8a	01 (1 byte)	INT32U	Grey/blue
allData	ab	08 (8 byte)	Boolean Val: TAG 83 Length 01 Data 00 Quality: TAG 84 Length 03 Data 030000	Light Yell.

Figure 48 GOOSE variable part TAGs

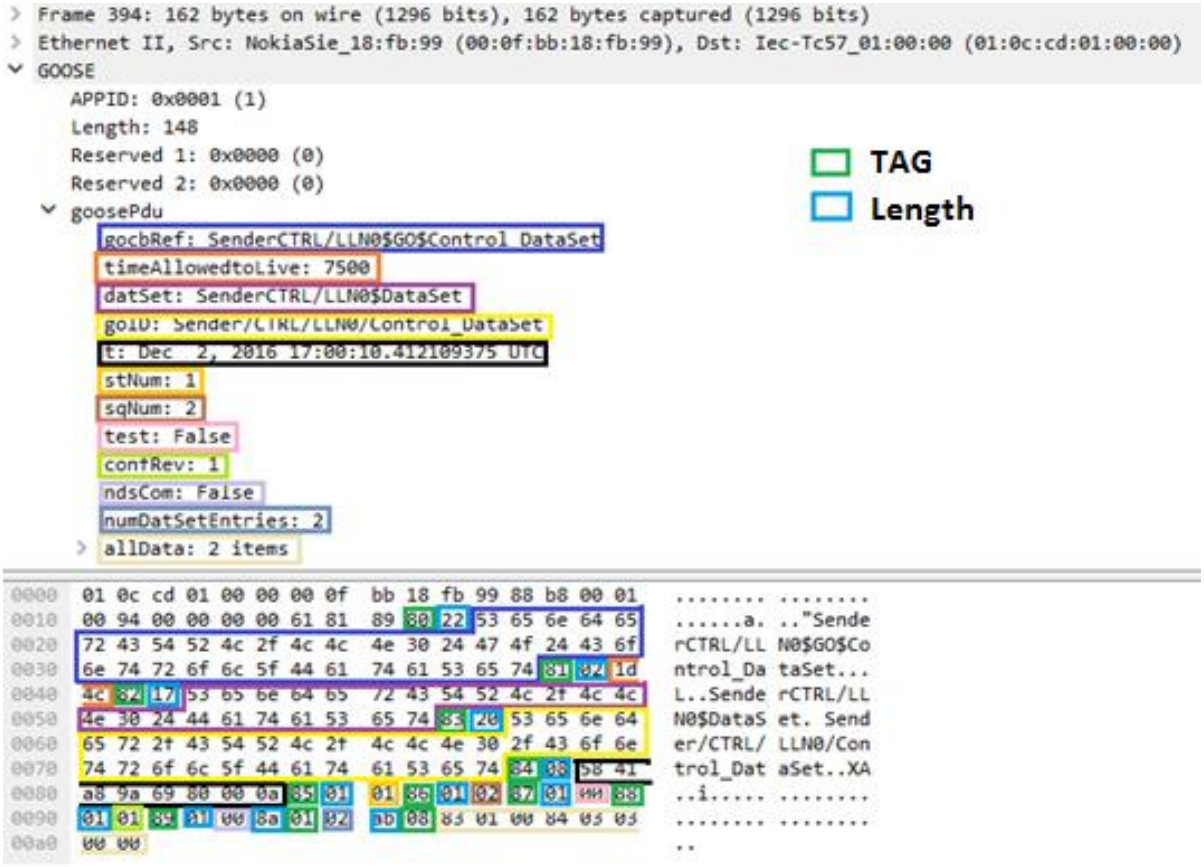


Figure 49 Captured GOOSE message detail

2.4.9 Redundancy protocols for IEC 61850

The IEC 61850 standard has become the backbone of substation automation, allowing for the first time interoperation between protection, measurement and control devices from different manufacturers on the same Ethernet local area network, station or process bus. This network is duplicated in substations that require a very high availability. Interoperability requires that all devices use the same redundancy concept.

IEC 61850 does not prescribe a topology, tree, star or ring, so any topology is conformant. It is even conformant to have the same physical Ethernet carrying both the station and the process bus traffic. For the station bus, the network topology that imposed itself in large substations is that each voltage level uses a ring of switches, which connect the IEDs, typically main protection, backup protection and control IEDs (Figure 50 left).

In large substations, the rings of the different voltage levels are connected in a tree form to the station level. The station bus therefore exhibits a mixed ring and tree topology. In small substations, for instance in medium voltage, there is typically only one IED per bay and each IED incorporates a switch element, such that the IEDs can be chained into a ring (Figure 50 right).

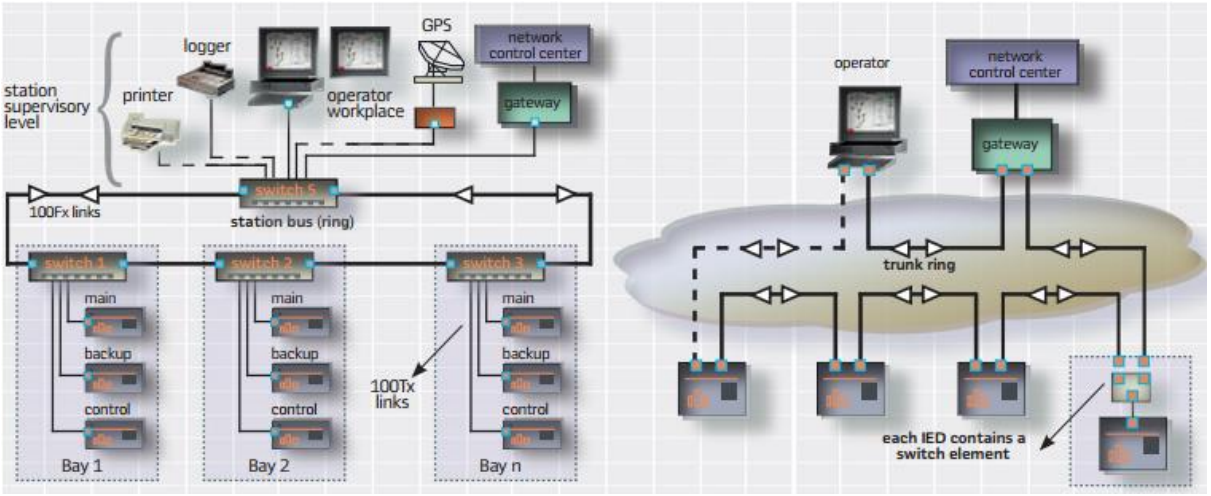


Figure 50 One level of station bus topology on the left and Ring with switching end nodes on the right

As the timing requirements for the station and for the process bus are distinct, they dictate the redundancy method to be used. When the station bus carries only command information, delays of some 100ms are tolerable, but when carrying interlocking, trip signals and reverse blocking, only a 2 milliseconds delay is tolerable in the normal case. Although it is unlikely that a failure will take place exactly when an (infrequent) control sequence is issued, no more than 4 milliseconds are tolerable in the worst case.

At the process bus level, IEDs are typically simple measurement and control devices connected in a tree form to a merging unit, which performs the protection functions and acts as interface to the station bus. The process bus carries real-time data from the measuring units, which requires a deterministic mode of operation, with maximum delays of 0ms. Here there is no difference between normal operation and worst case.

In accordance with what stated above, Figure 51 shows the timing requirements that have been compiled by the TC57 Working Group 10.

Communication Partners	Service	Bus	Recovery Time
SCADA to IED – Client-Server	IEC 61850-8-1	Station Bus	400ms
IED to IED – Interlocking	IEC 61850-8-1	Station Bus	4ms
IED to IED – Reverse Blocking	IEC 61850-8-1	Station Bus	4ms
Bus bar protection	IEC 61850-9-2	Station Bus	0ms
Sampled Values	IEC 61850-9-2	Process Bus	0ms

Figure 51 Timing requirements for IEC 61850

For the station bus of small substations that operate with a single ring and a limited number of IEDs, a “redundancy in the network” solution such as Rapid Spanning Tree Protocol (RSTP) is applicable, as long as the station bus does not carry interlocking information. Therefore, this solution has only limited applicability and it is not further detailed here. By contrast, IEC 61850 specifies a network redundancy that fulfils the requirements of substation automation, for the station bus as well as for the process bus. It is based on two complementary protocols defined in the IEC 62439-3 standard: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR) protocol.

In both protocols, each node has two identical Ethernet ports for one network connection. They rely on the duplication of all transmitted information and provide zero-switchover time if links or switches fail, thus fulfilling all the difficult real-time requirements of substation automation.

2.4.9.1 PARALLEL REDUNDANCY PROTOCOL

Zero recovery time for data transmission can occur when the data appears on multiple paths simultaneously. The multiple paths of PRP are two redundant networks. These networks must be completely independent networks. The networks use standard Ethernet switches, with both managed and unmanaged switches. The general concept of PRP is illustrated in Figure 52.

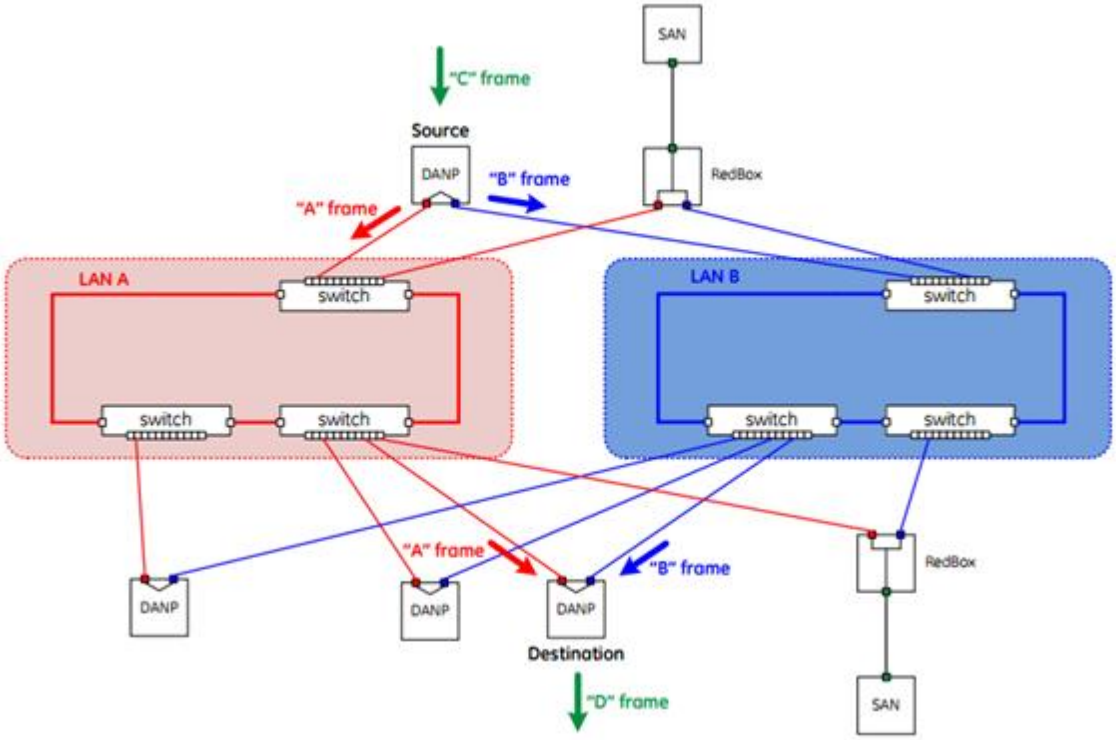


Figure 52 PRP concept

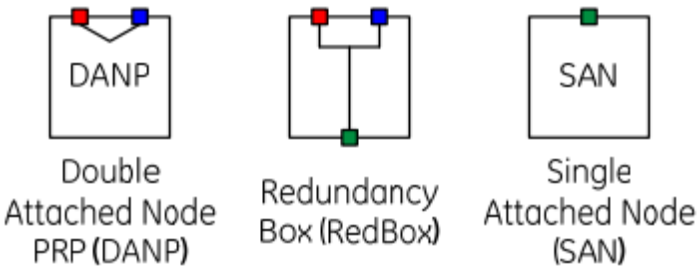


Figure 53 PRP devices

A source device publishes a PRP “C” frame that is then mirrored as the PRP “A” frame to LAN A, and as the PRP “B” frame to LAN B. A subscribing device, connected to both LANs, accepts the first frame received, and discards the second frame received. The resulting output is the PRP “D” frame. If LAN A experiences a failure, the “A” frame will not be received in a timely fashion by the destination device. However, the “B” frame will still be received as normal.

As illustrated in Figure 53, there are several different types of nodes that can attach to a PRP network, below is a list of such nodes:

- DANP: It is a “double attached node implementing PRP”. A DANP has two ports (port A and port B) that have the same abilities, and in particular, it could be used alternatively if only one LAN would be connected. A source DANP sends the same frame over both LANs. A destination DANP receives the mirrored frames from both LANs within a certain time, consumes the first frame and discards the duplicate. The “C” frame and the “D” frame are internal to the DANP.
- SAN: it is a “single attached node” that has only one port for the purpose of this protocol, so that no special requirements apply.
- Redbox: It is used to attach SANs to a PRP network. The Redbox acts like a DANP on the PRP side. The Redbox mirrors the “C” frame published by a SAN, and creates the “D” frame from mirrored PRP messages to send to a SAN.

Figure 54 shows the actual Ethernet frame when the PRP feature is implemented. PRP uses the Ethernet frames, and these PRP Ethernet frames are intended to be compatible with standard LAN switches as the Redundancy Control Trailer (RTC) is embedded inside the Ethernet PDU. The switches simply need to support oversize frames of up to 1528 octets.

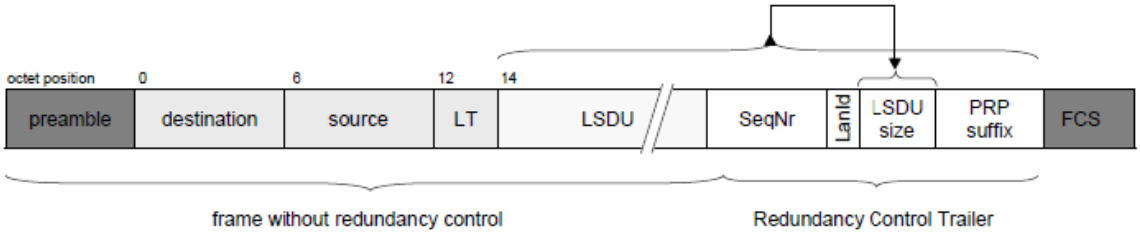


Figure 54 PRP frame

There are many benefits to using PRP for high-availability networks. The first of these is that PRP explicitly achieves “zero” fail over time, due to the use of mirroring frames across both networks.

Another advantage is that the PRP networks can use any topology: star networks, ring networks, and any other connection. And these networks are built using standard LAN switches. Traditional devices can still be connected to these networks individually.

Most importantly for GOOSE messaging and sampled values, it is still possible for traffic shaping through VLANs, message priority, and MAC address filtering.

2.4.9.2 HIGH-AVAILABILITY SEAMLESS REDUNDANCY

HSR uses a different method to provide multiple paths for data. All devices are connected to the network in a ring configuration, as illustrated in Figure 55.

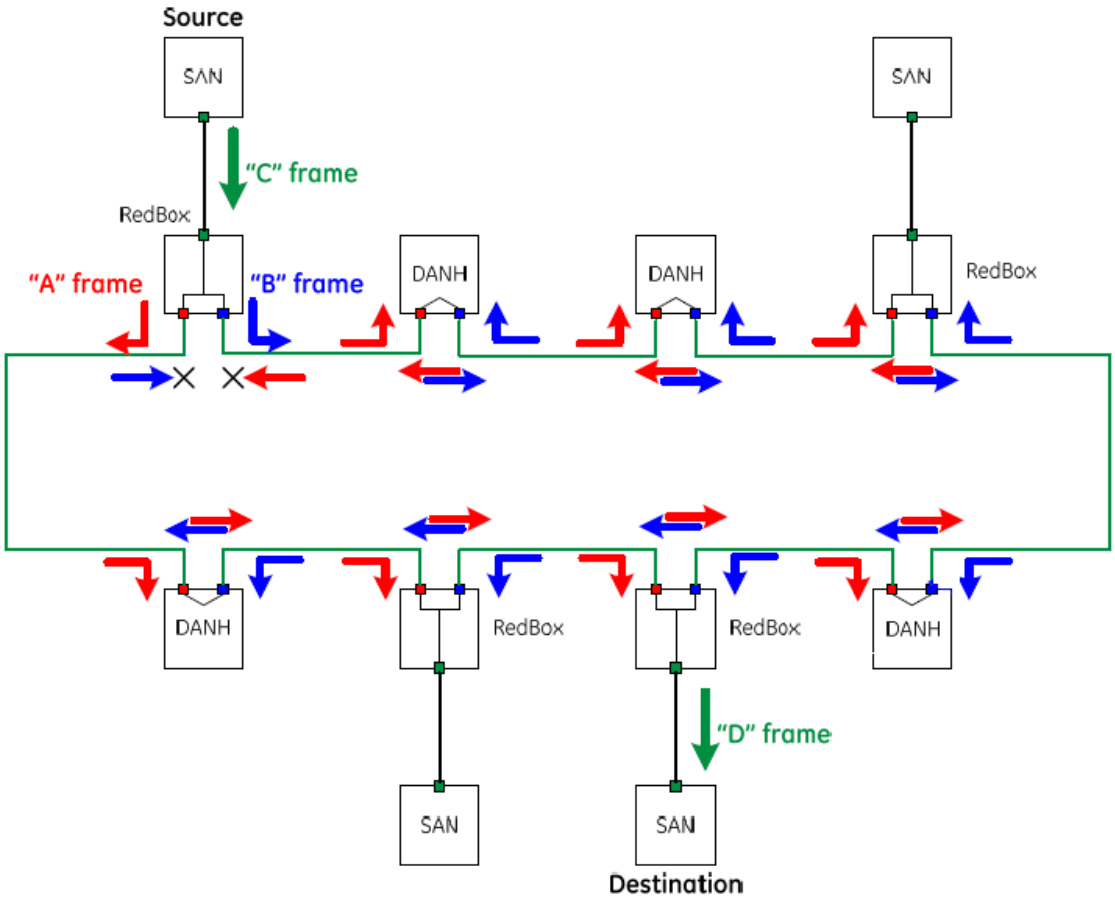


Figure 55 HSR concept

A source device publishes identical frames, the “A” frame and the “B” frame, in opposite directions out of the two ports. A destination device receives two identical frames on each port within a certain interval. The device uses the first frame received, and discards the second frame. If a network link fails, only one frame is received, and this frame is used. Even with a large number of nodes on the network, the time difference between the receipt of the two frames is negligible, so zero recovery time is achieved.

The nodes support the IEEE 802.1D bridge functionality and forward frames from one port to the other, except if the node has already sent the same frame in that same direction. To keep traffic from continually passing around the ring, a node will not forward a frame that this specific node injected into the ring.

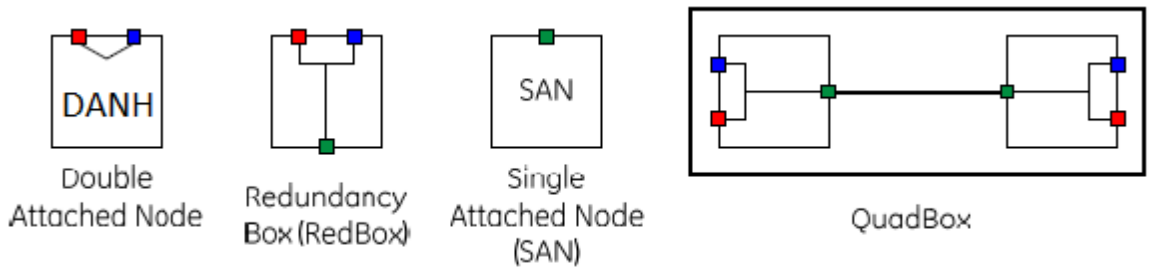


Figure 56 HSR devices

As illustrated in Figure 56, there are several different types of nodes that can attach to an HRS network, below is a list of such nodes:

- DANH: It is a Double Attached Node implementing HSR. A DANH has two ports (port A and port B) that have the same abilities.
- SAN: Single Attached Nodes are not directly supported in HSR; they must always connect via a Redbox. As LAN switches are considered SANs under IEC62439-3, LAN switches are not supported in HSR rings.
- Redbox: It is used to attach SANs to an HSR network. The Redbox acts like a DANH on the HSR side. The Redbox mirrors the “C” frame published by a SAN, and creates the “D” frame from mirrored HSR messages to send to a SAN.
- QuadBox: HSR also introduces the concept of a QuadBox, a quadruple port device that connects two peer HSR rings. The QuadBox behaves as an HSR node in each ring, and can filter the traffic between rings and forward traffic from ring to ring.

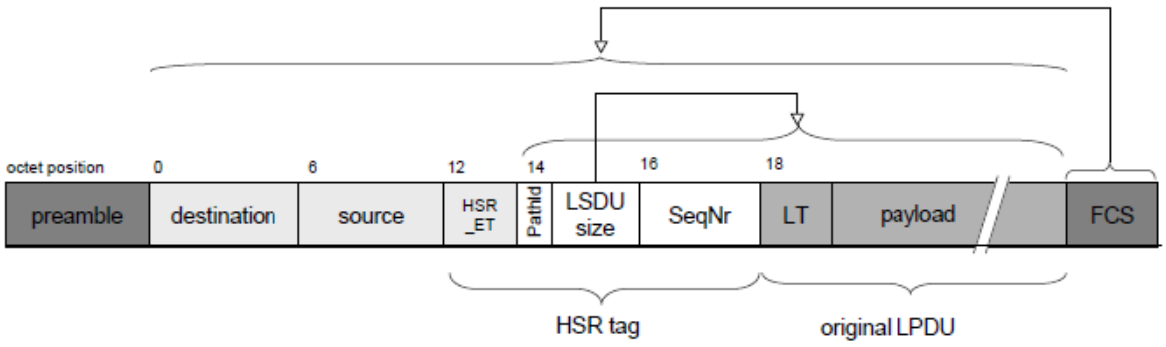


Figure 57 HSR frame

Figure 57 shows the HSR frame structure; as the HSR tag is not inserted into the payload as with PRP frames, HSR Ethernet frames are not compatible with standard Ethernet frames. This means the only frames possible on an HSR ring are HSR frames. The HSR tag includes an EtherType HSR identifier, a path identifier, the frame size, and a frame sequence number. The frame sequence number, combined with the source address in the frame, is used to identify duplicated frames.

While HSR nodes do support VLAN traffic, it is not practical to do traffic shaping using VLANs on an HSR ring. Every source data frame is published in both directions around the ring. For HSR to provide high availability, both unicast frames must travel completely around the ring to destination devices.

In case of multicast frames (GOOSE message for example), both frames must travel completely around the ring until reaching the originator devices (i.e. the “whole ring”). Therefore, any VLAN will have to include every node on the ring. This means that on an individual HSR ring, no traffic shaping is possible except when applied to C or D frames which are outside of the HSR ring.

Figure 58 shows the actual GOOSE message structure when the two redundancy protocols are implemented[15].

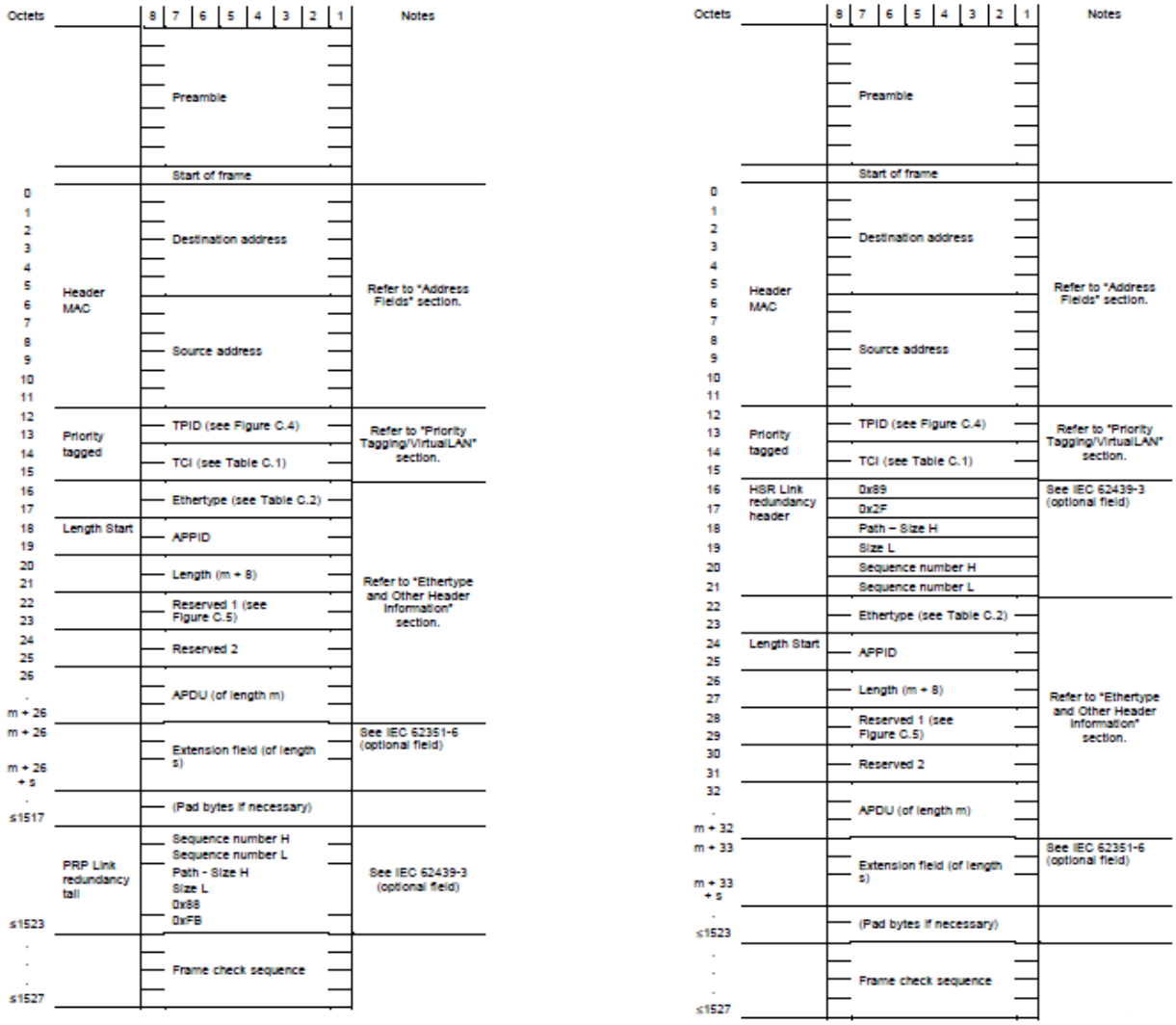


Figure 58 GOOSE message using PRP on the left and HSR on the right

2.5 ARCHITECTURES AND RESULTS

2.5.1 The network under test

The implemented network contains all the components that typically constitute a system with the main automation functions of process control, power management and general-purpose applications:

- Profinet network: PLC connected to several Profinet IO devices through a Switch;
- Intelligent Electrical Device interfaced via IEC 61850;
- TCP/IP traffic generated with a traffic generator;

- SCADA: Supervision, Control and Data Acquisition.

The Profinet network used consists in a portable demo kit, property of GFCC (Genoa Fieldbus Competence Centre) that is a spin-off of the University of Genoa and a Profibus and Profinet certificated competence centre. The kit is made with components from various manufacturers in order to show the interoperability of the Profinet standard. The Profinet network consists in a Siemens S7-300 PLC with a Profinet interface card, a Siemens Scalance X208 switch and 4 I/O Devices from Beckhoff, Phoenix, Siemens and Wago.

The SCADA is Zenon from COPADATA: it can be interfaced both with a Profinet network and with a 61850 network. A soft-PLC named Straton is integrated in the SCADA and it can control both a Profinet and a 61850 network. This allows to create logics where the 2 protocols can interact each other. The mentioned software runs on a PC.

Another PC will be used in order to simulate Ethernet TCP/IP traffic using an open source packet crafter and traffic generator called Ostinato.

The devices interfaced with IEC 61850 are a telecontrol unit provided by Wago, with three I/O modules and a Siemens Siprotec 7SJ62 protection relay.

The network under test and all the devices used in this work are shown in Figure 59.

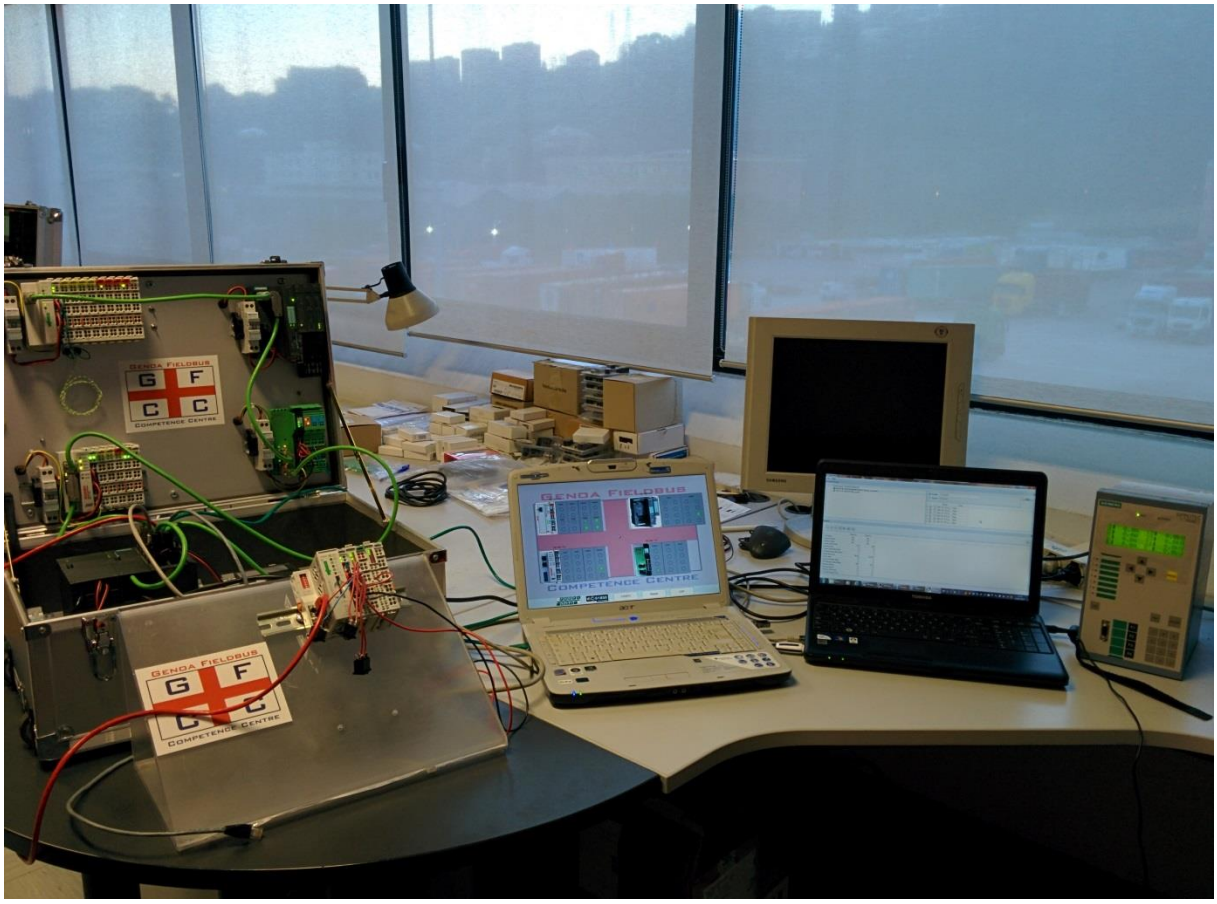


Figure 59 The network under test

2.5.2 Architectures

Scope of the tests is to verify the fitness of different architectures of the automation network for different process applications. A general model of a process control system is based on a control room and one or more areas where different sections of the process operate (Figure 60).

For the purpose of this test, and with a good approximation to common industrial realities, in each area is present a piece of the process, a part of the electrical distribution system and some general purpose applications such as, one or more security cameras. To implement the required control and supervision functions we assume that:

- the process control is based on intelligent field devices (IFD) and remote I/O devices connected via Profinet;

- the supervisory, protection, and control functions for the electrical distribution system are actuated by means of intelligent electrical devices (IED) connected via IEC 61850;
- the “general purpose” functions are implemented by devices connected via Ethernet TCP/IP.

These three protocols are implemented in each area of the plant. The automation functions may be either implemented locally, e.g. using PLCs, or centralized in the Control Room. In any case it is mandatory to have a complete access to all the sub-systems from the Control Room.

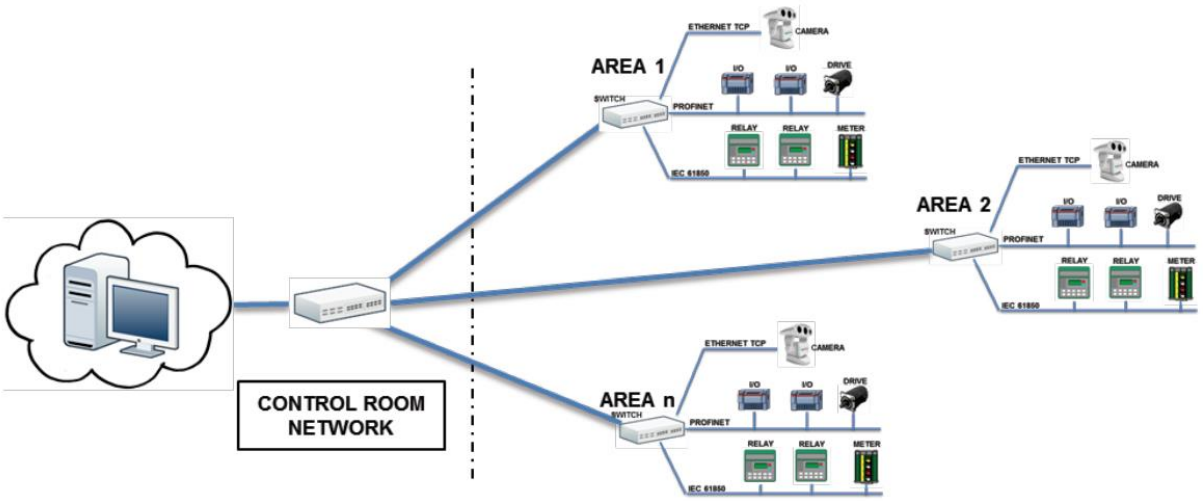


Figure 60 Typical plant structure

Three different architectures are presented to satisfy different automation requirements.

The first architecture, named “Distributed automation architecture”, is especially suited for processes that require local control in the various areas, e.g. manufacturing, where each machine or island has its own controller. In these processes the control requirements are normally very strict in field (control cycles of few milliseconds), so local controllers are mandatory. The Control Room accesses the process controllers for supervisory functions, and the same happens with the electrical distribution and the security cameras (that may also have safety functions in some processes). This typical

architecture separates the control functions from the supervisory functions that can run independently on different machines.

Figure 61 shows the tested architecture for these goals. The automation functions are actuated by the PLC of each area that is connected to the remote I/Os and intelligent devices via Profinet. The PLC configures the PNIO network, assigns the IP addresses to the connected devices, and configures them. The program executed by the PLC commands the outputs of the devices with the required cycle time. The PNIO network is connected to the PC through the switch, where also the IEC 61850 and TCP/IP sections enter. For simplicity sake, Figure 61 shows a single PC in the Control Room, but it can include a network of PCs, database or application servers, thin clients, etc.

All the devices are connected to the same network, but there is no operating interaction between the Profinet IO and the IEC 61850 devices. The unique common point is the SCADA that has the drivers for both protocols. The HMI and the alarm management system in the Control Room are transparent to the used protocols. For the configuration of the IEC 61850 IEDs it is necessary to install in at least one of the computers in control room the specific software supplied by the producer together with the IEDs. This computer becomes the Engineering Work-Station for the IEC 61850 network. The same happens for the video cameras that have a specific software tool for configuration, setting, recording, display, etc.

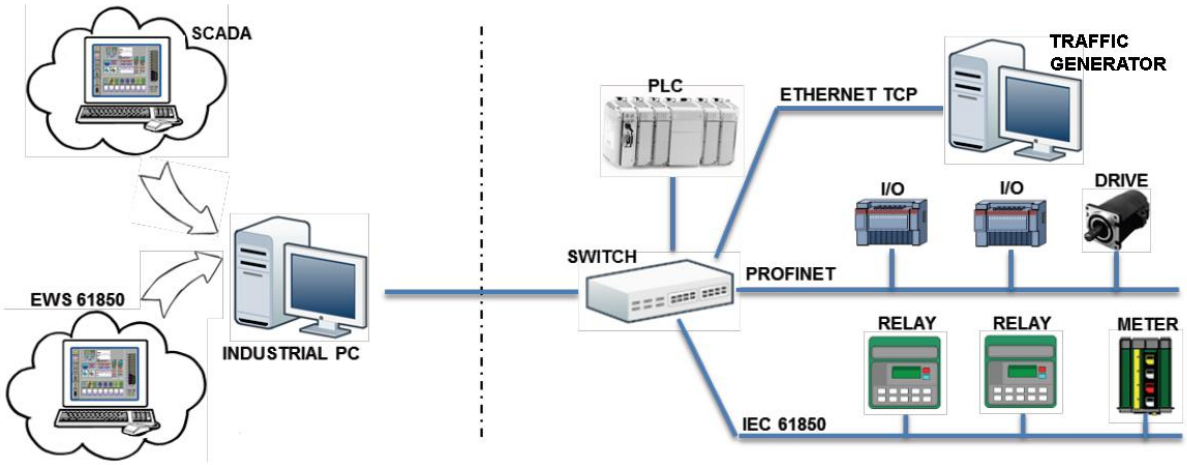


Figure 61 Architecture for Distributed Automation

When the process does not require a controller in each area, it is possible to use the Ethernet network for collecting data that are centralized in the Control Room where a unique PLC controls the whole plant. Such a configuration is feasible if:

- required system availability is low,
- no safety function is necessary,
- maintenance outages for the whole plant are possible,
- control cycle is in the order of tenths or hundreds of milliseconds,
- the number of variables from/to the field is limited,
- there is no functional interaction between the process and the electrical distribution.

The main advantage of this architecture, called “Centralized automation architecture”, is the reduction of the number of PLCs that means reduced installation costs and ease of software maintenance. The SCADA and EWS functions are identical to those of the architecture for Distributed Automation.

A better and more powerful architecture is presented in Figure 62. This architecture is very simple and it minimizes the amount of required hardware. A soft-PLC replaces the PLC and it runs on the same PC where the SCADA runs. This solution allows a very effective integration of process data and electrical distribution data to create common logics and functions. The Soft-PLC has drivers for all the protocols implemented on the Ethernet, so a seamless communication data exchange is possible with all the subsystem of the plant. Functional logics that requires data or command from or to the process and the electrical distribution can be easily implemented with this architecture[4][5].

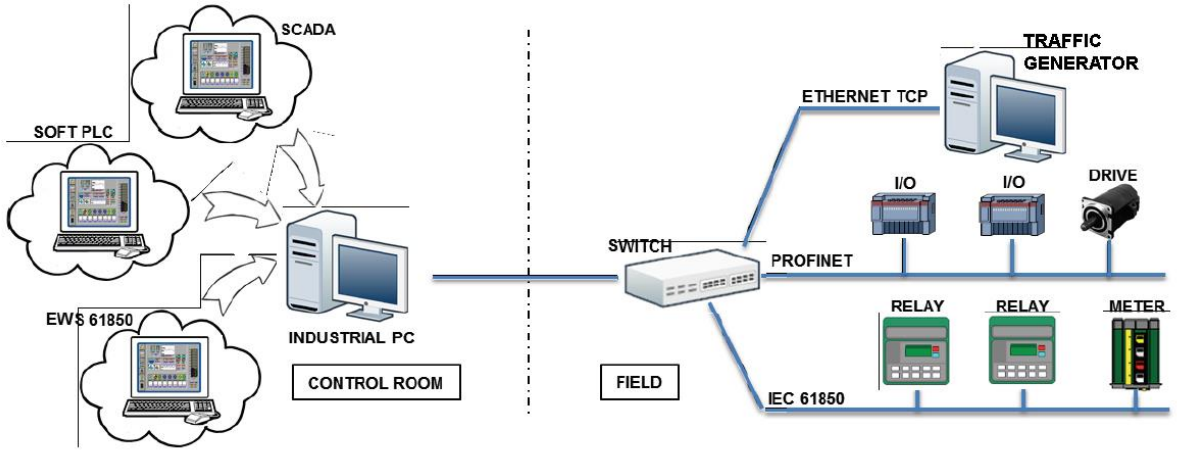


Figure 62 Centralized PC-based automation architecture

2.5.3 Test procedures

Since the network consists of three different communication protocols, the aim of this section is to verify that using different protocols on the same physical bus doesn't disturb the time critical performance of Profinet and IEC 61850, and, if they are disturbed, where is an approximate limit of usage. This is the reason why the focus is on comparing the performances measured when the load on network is low with the performances when the traffic on the network is high.

Both Profinet IO and IEC 61850 need strict timing characteristics to fulfil their roles. For this reason, to measure the performance of the network time-related performance indicators are needed. Typically, these indicators are latency and jitter.

In general, the latency is defined as the time delay between the cause and the effect of some physical change in the system being observed. In this case, the latency is measured as the time spent between two sequential frames. Since two different protocols runs on the same physical network, it is important to suite the concept of latency for Profinet and IEC 61850.

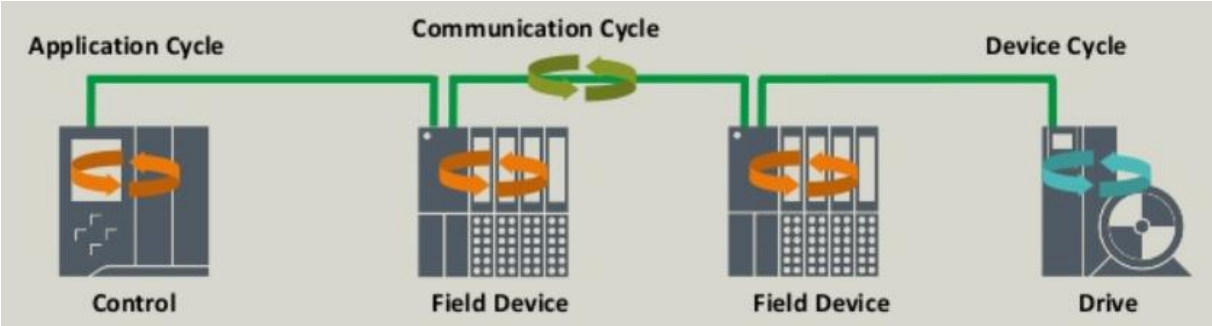


Figure 63 Communication cycle time and device cycle time

The time performances of the Profinet network are measured in term of Cycle Time and Jitter. Profinet IO's data exchange is based on the cyclic communication between the controller and the field devices. During the configuration of the network, the cycle time is set (if required) with a different value for each IO-device connected to the network. This means that the data exchange between the controller and the device cyclically repeats every x ms. The device's cycle time is the product of the reduction ratio and the Send Clock Factor X 31.25µs. For example: to set the cycle time of the device to 4 ms, the Send Clock Factor can be set to 32 and the Reduction Ratio to 4. The cycle time of all the Profinet devices in the network will be evaluated with five different amount of TCP/IP traffic[4][6][7][8][9].

Jitter measures the variations of the Cycle Time, and it can be defined by the formula:

$J=|CT-CT_m|/CT$

Where: CT rated cycle time

CT_m measured cycle time

Deviations of the CT may be caused by:

- random transmission delays,
- transmission errors,
- latencies in switches or any other network node,
- excessive usage of the bandwidth by Non Real Time communication.

Jitter can be also computed after measuring a series of consecutive CTs and calculating the standard deviation of the mentioned series.

The value calculated adding two times the jitter to the mean measured cycle time is the max value bigger than the 95% of the measured cycle times.

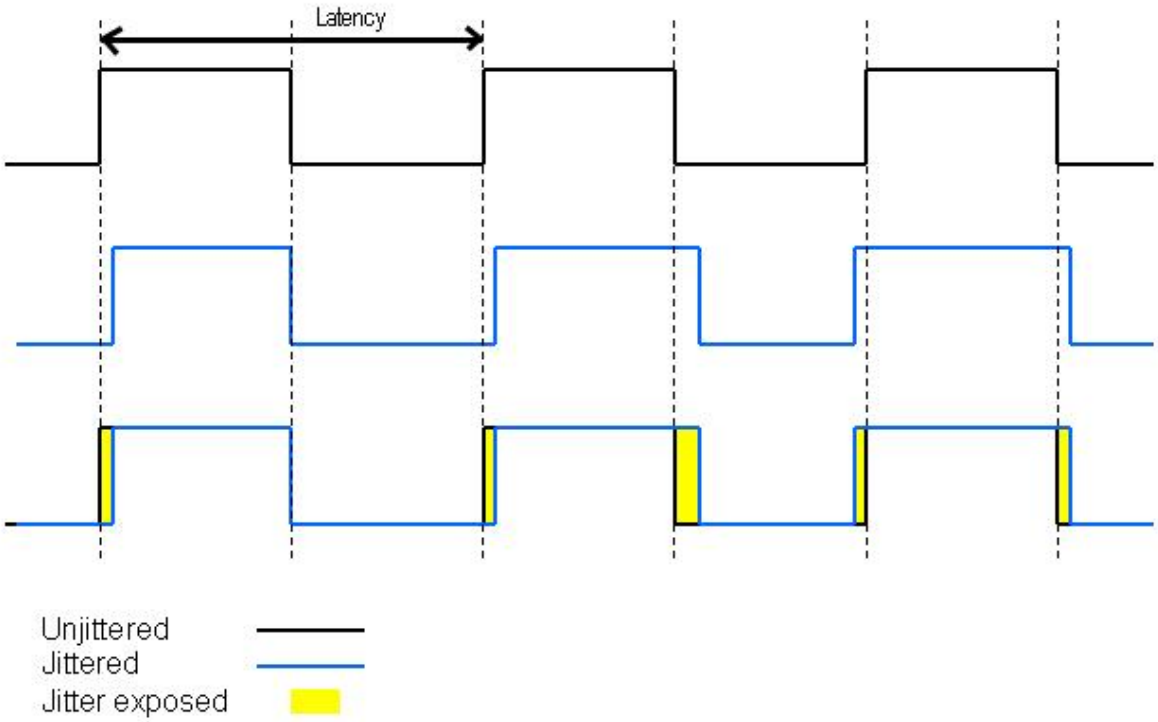


Figure 64 Jitter

The “Transfer Time” is defined in IEC 61850-5 as the time required for delivering a process value from a sending physical device to the process logic of a receiving device and it can be considered as the sum of three time intervals:

- t_a: the time required for the sending device to transmit the process value,
- t_b: the time required for the network to deliver the message,
- t_c: the time required for the receiving device to deliver the value to its process logic.

While IEC 61850-10 focuses on t_a and t_c to measure the performances of single IEDs, the aim of this paper is to test the network performances, so t_b is the KPI most useful

for the purposes of the tests. Unfortunately, it is not possible to clearly separate and measure t_b from the whole Transfer Time. As experiences seem to prove, t_a and t_c are considered constant as the traffic on the network grows, and only t_b is considered variable.

The tests consider two services of the IEC 61850 protocol, the service Get/Set DataValues and the GOOSE messages. A 61850 client uses the service Get/Set DataValues for reading or writing cyclically the variables of a server. The update time of the Get Data service is named “polling rate” and it can be set during the configuration of the IEC 61850 Client (e.g. the SCADA). The indicator considered for the service GetDataValues is the delay between two sequential readings of the same variable.

Commands to a server are sent as standard request frames. Once the server has completed the task requested by the client, it sends back a response frame to confirm the success of the operation. The delay between the request and response frames is the PI measured for SetDataValues service.

GOOSE communication does not consist in an exchange of a request and a response. For this reason, it is not possible to define an indicator that measures the influence of different network traffic on the transmission of GOOSE messages.

Wireshark is the software network analyser used to measure the performance of the networks under test. Software network analysers are intrinsically inaccurate due to the fact they run on a non real time operating system; the time performances are affected by the operating system latency and jitter. Several tests about software network analysers found that the error introduced by using traditional software sniffers are in the order of several µs. Since the time between two sequential frames on the networks under test goes from a minimum of 4 ms to a maximum of 128 ms, an error in the order of 10^{-3} is considered acceptable. Furthermore, the goal of this test is to verify where is the limit of usage of the available bandwidth in order to prevent process automation and power system automation failures. This is the reason why a more accurate but more expensive measurement method is not considered.

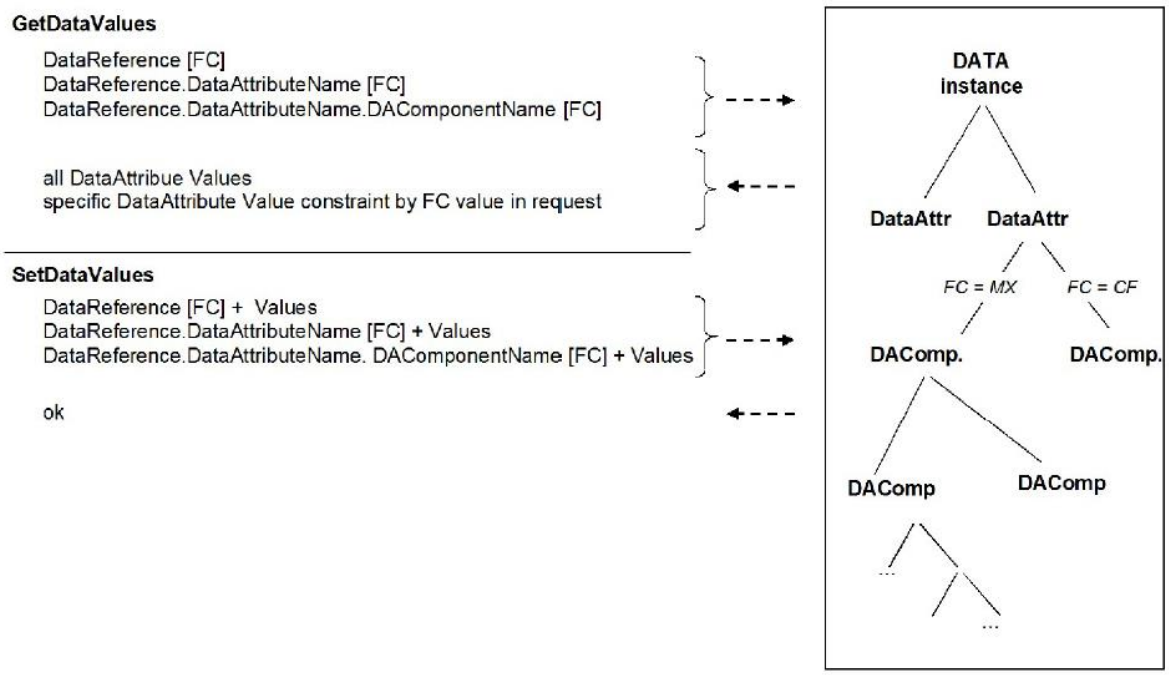


Figure 65 GetDataValues and SetDataValues services

2.5.4 Test Setup

The two physical layouts of the Network under Test are shown in the figures below.

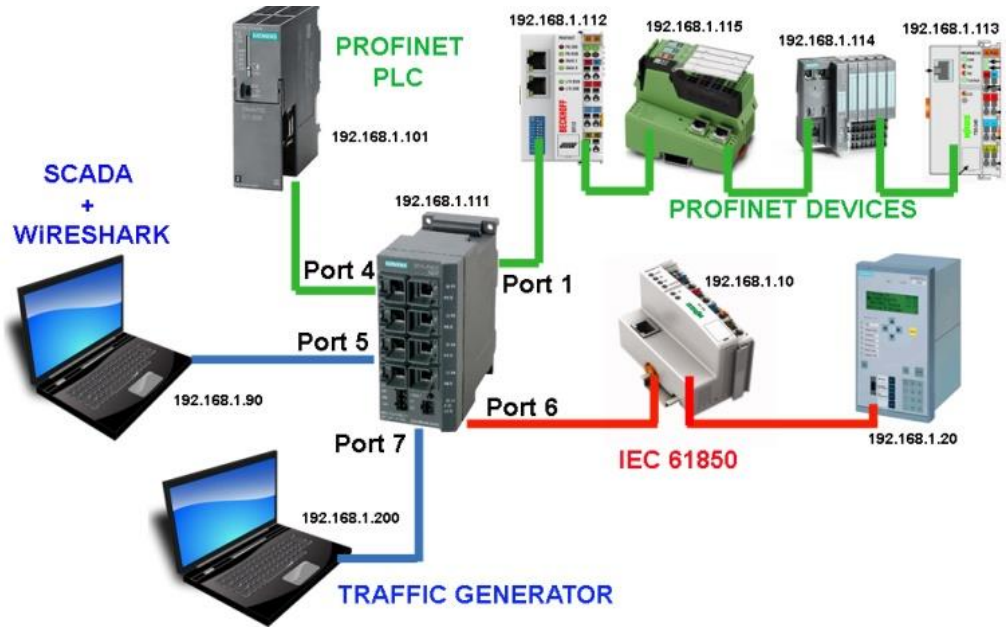


Figure 66 Distributed automation architecture

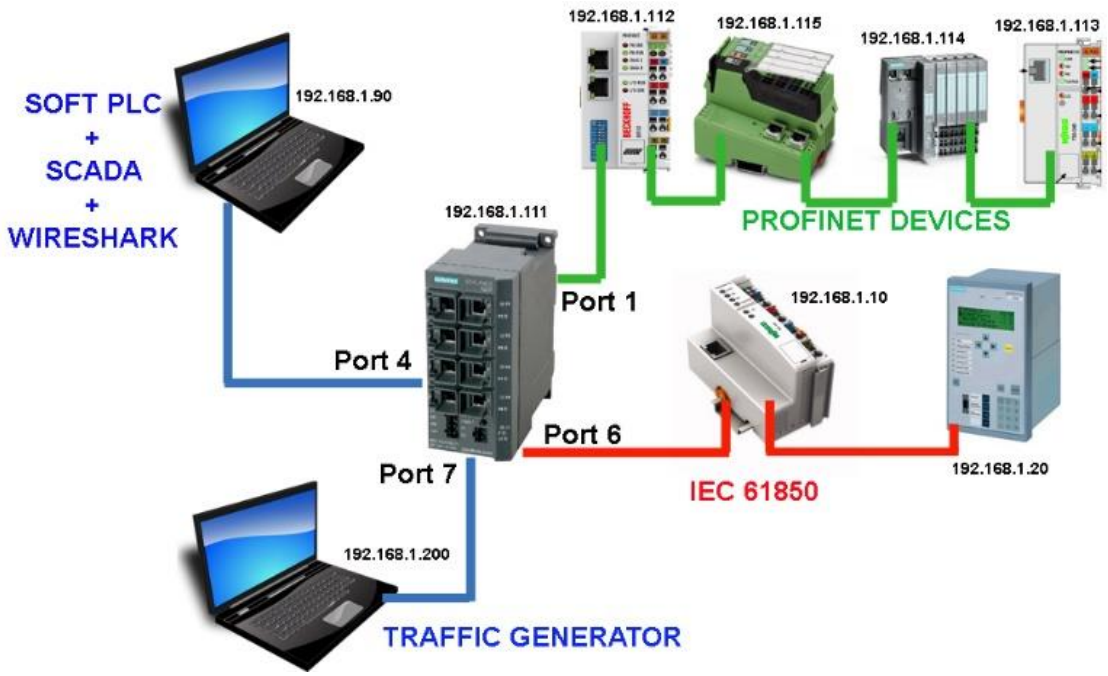


Figure 67 PC-based automation architecture

Two PCs are connected to the NUT. The SCADA runs on the first, and the traffic generator runs on the second. A packet analyser running on the PC used for the SCADA (Wireshark) measures the time indicators.

The architecture of Figure 66 is a representation of a standard distributed architecture described before. The layout of Figure 67 is a model of a PC-based architecture where the control (Soft-PLC) and supervision (SCADA) run on the same machine.

In order to simulate the data exchange of a real network, the Profinet IO network runs a simulated logic where the outputs of the remote IOs are cyclically changed.

The IEC 61850 communication is established between the SCADA and the IEC 61850 servers. The protection relay is configured to produce a GOOSE message every 2 seconds.

Using the traffic generator, five different network scenarios are created, from 0% TCP/IP traffic on the network up to 90%. These frames are composed of 1518 bytes, and the MAC address is configured for broadcast messages. The frames are gathered in streams, and each stream lasts for 10 seconds. A packet analyser captures all the

frames that pass through the mirrored port of the switch and the recorded time stamps are used to evaluate the time performances.

2.5.5 Test results

Table 1 Distributed architecture

	Delay between two sequential frames [ms]					
	192.168.1.112 (32 ms)			192.168.1.115 (128 ms)		
	Min	Max	Avg	Min	Max	Avg
0%	31.42	32.72	32.01	127.85	128.13	127.98
20%	31.57	32.45	32.01	127.74	128.25	127.98
40%	31.49	64.02	32.03	127.70	128.28	127.98
60%	31.46	128.25	32.27	127.72	255.93	128.60
80%	31.75	160.19	32.35	127.68	383.91	129.24
90%	31.45	160.01	32.47	127.70	384.15	129.88
	Delay between two sequential frames [ms]					
	192.168.1.114 (2 ms)			192.168.1.113 (4 ms)		
	Min	Max	Avg	Min	Max	Avg
0%	0.78	3.21	2.00	2.01	6.03	4.00
20%	0.98	3.10	2.00	3.15	11.77	4.02
40%	0.84	13.97	2.00	fail	fail	fail
60%	1.15	64.00	2.02	fail	fail	fail
80%	1.54	139.86	2.02	fail	fail	fail
90%	0.97	134.21	2.03	fail	fail	fail

The table shows that the 19.168.1.113 device stops to work when the TCP/IP traffic is more than 20%. This is probably because its CPU is not able to manage the frames those arrives at its interface while keep working with Profinet Real Time data exchange. From the packets captured with the packet analyser it is possible to notice that after a few cycles, the Application Relation between the controller and the device is interrupted. The other devices on the network works with no disturbances since the traffic is more than 40% of the bandwidth, even if the average delay is quite constant.

Table 2 shows a similar result also for PC-based architecture, but 112, 114 and 115 devices keep working with a traffic of Mbps. A comparison of calculated jitter reveals that PC-based architecture is more stable.

Table 2 PC-based architecture

	Delay between two sequential frames [ms]					
	192.168.1.112 [32 ms]			192.168.1.115 [128 ms]		
	Min	Max	Avg	Min	Max	Avg
0%	31.61	32.39	32.01	127.79	128.19	127.98
20%	31.50	32.49	32.01	126.44	129.37	127.98
40%	31.60	32.38	32.01	127.75	128.20	127.98
60%	31.81	32.21	32.01	127.80	128.15	127.97
80%	31.51	63.90	32.03	127.76	255.84	128.29
90%	30.39	33.63	32.01	127.80	128.15	127.98
	Delay between two sequential frames[ms]					
	192.168.1.114 [2 ms]			192.168.1.113 [4 ms]		
	Min	Max	Avg	Min	Max	Avg
0%	0.01	4.54	2.00	2.25	5.78	4.00
20%	0.10	3.80	2.00	2.46	12.18	4.01
40%	0.82	3.19	2.00	fail	Fail	fail
60%	0.61	3.33	2.00	fail	Fail	fail
80%	1.04	14.05	2.00	fail	Fail	fail
90%	0.77	3.21	2.00	fail	Fail	fail

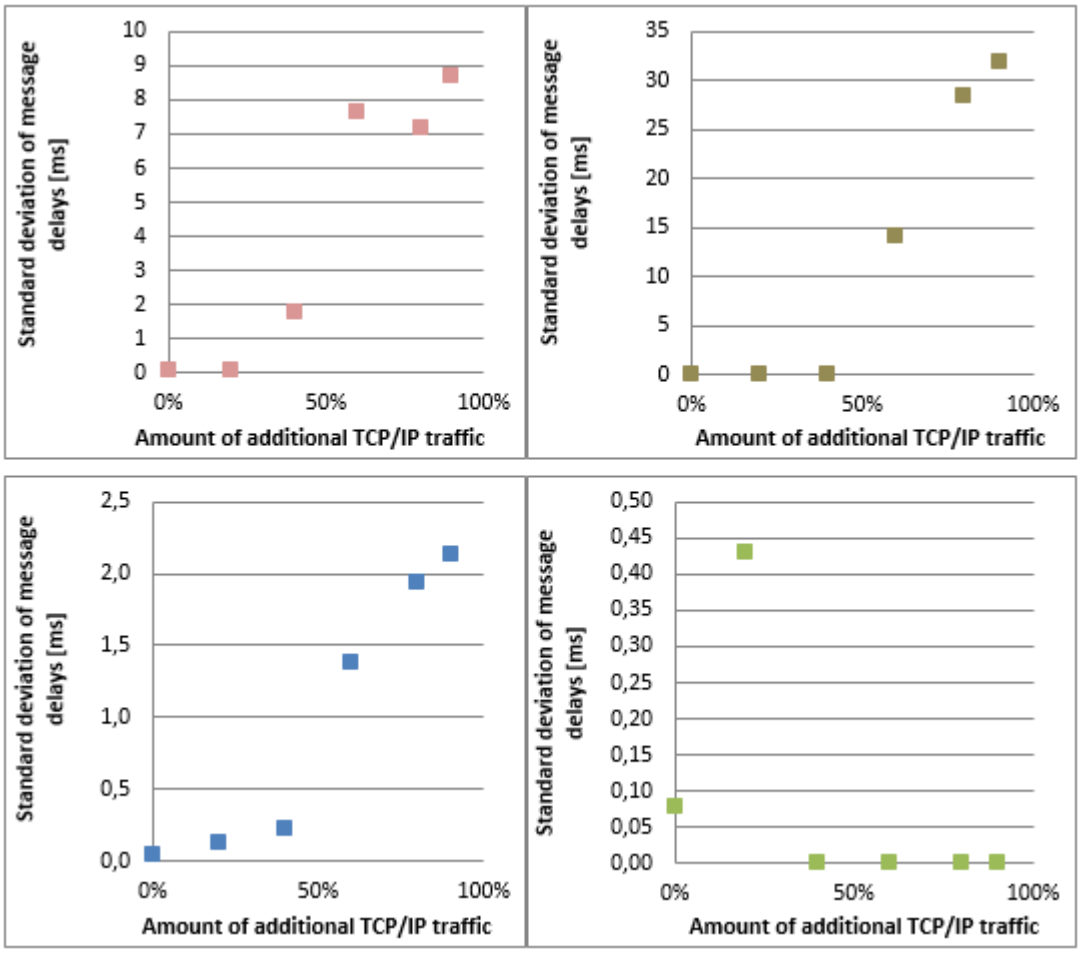


Figure 68 Jitter vs amount of TCP/IP traffic: distributed architecture

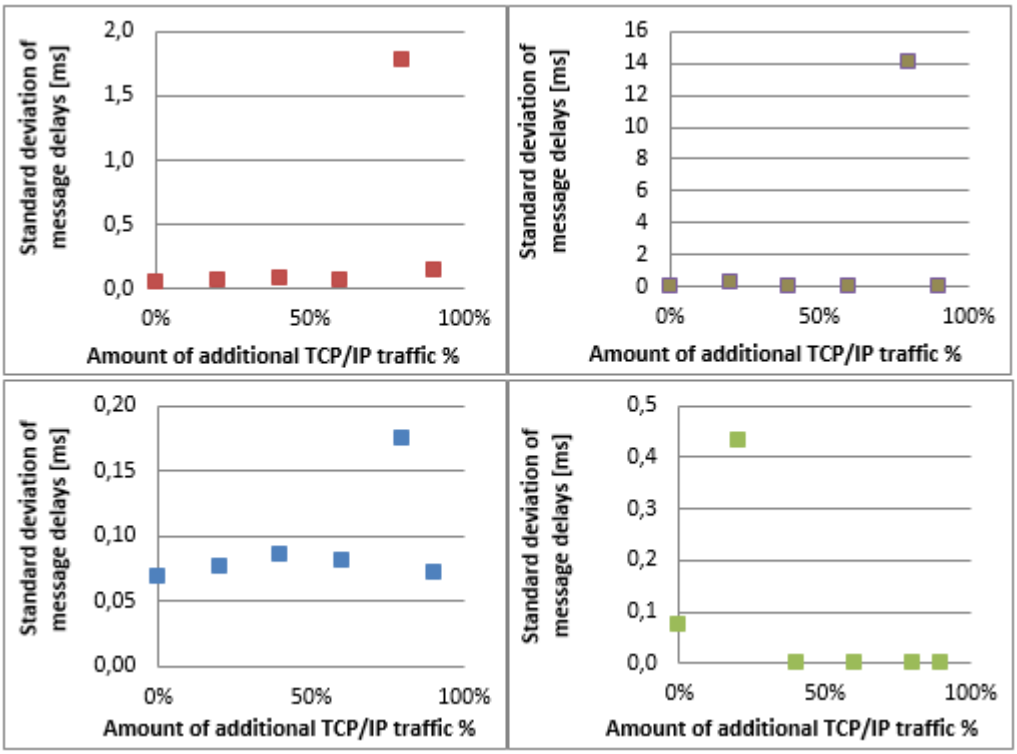


Figure 69 Jitter vs amount of TCP/IP traffic: PC-based architecture

Figure 68 shows that also the standard deviation of the measured delay increases when the TCP/IP traffic is more than 40Mbps.

Figure 70 represents the trend of the cycle times of the Profinet IO devices for the five different traffic scenarios and for the two different architectures observed.

It is apparent that for the distributed architecture the cycle time starts to grow significantly when the traffic on the network is higher than 40% of the bandwidth. One of the devices interrupts its data exchange with a traffic higher than 20% of the bandwidth. In the PC-based architecture, the remote I/O device at 192.168.1.113 address responds to the traffic on the network like in the distributed architecture. On the contrary, the other IO devices continue working without losing performances with a traffic on network up to 60% of the bandwidth.

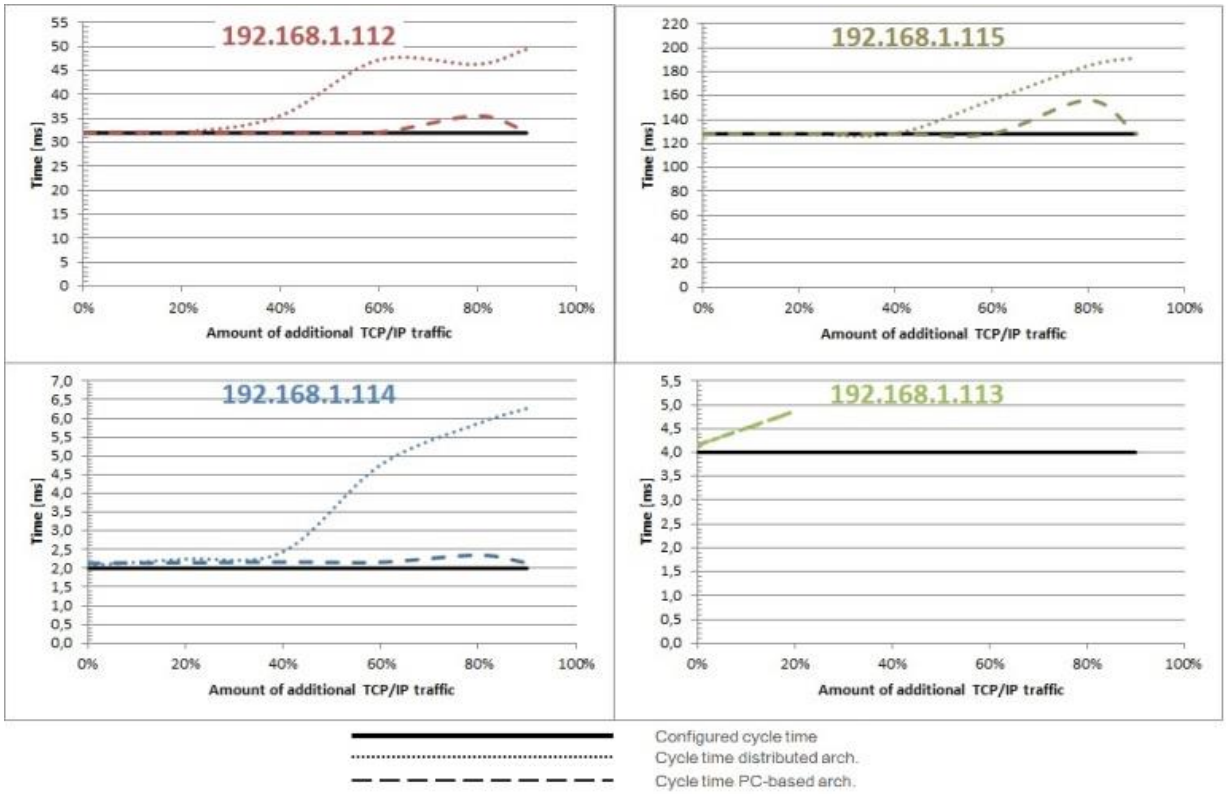


Figure 70 Profinet IO network time performance comparison

The time parameters measured for IEC 61850 are reported in Table 3.

With this configuration the polling rate of the variable exchange between client and server is set to 2 seconds.

It is possible to see that under 60% of load on the network the polling time is constant and the average of the measured value is almost 2 seconds. With a traffic of 60 Mbps the maximum value measured is more than four times the configured polling rate. When the bandwidth is almost full occupied the communication between the device and the SCADA is interrupted, probably due to an overload of the device itself.

The time between the request and the response of the SetDataValue service is represented in the second column of Table 4. Since it is not possible to “a priori” configure the delay of this mechanism, we can only analyse that more TCP/IP traffic on the network takes to a longer request-response exchange.

Table 3 Time performance for IEC 61850 distributed architecture

	GetDataValues [s]		SetDataValues [ms]	
	Max	Average	Max	Average
0%	2.0131	2.0124	17.82	12.58
20%	2.0592	2.0124	19.92	13.58
40%	2.0139	2.0124	32.73	21.75
60%	9.0012	3.8120	975.82	419.30
80%	Fail	Fail	Fail	Fail
90%	Fail	Fail	Fail	Fail

For PC-based architecture, the polling rate of the variable exchange between the SCADA and the IEC 61850 server is set to 1 second.

Table 4 Time performance for IEC 61850 PC-based architecture

	GetDataValues [s]		SetDataValues [ms]	
	Max	Average	Max	Average
0%	1.0302	1.0118	18.93	13.15
20%	1.0912	1.0096	20.98	12.89
40%	1.0883	1.0044	76.29	29.05
60%	4.931279	2.560799	1775.72	1273.90
80%	Fail	Fail	Fail	Fail
90%	Fail	Fail	Fail	Fail

The measurements show that the time performances remain almost constant until 40Mbps of TCP/IP traffic. At 60% of bandwidth usage the delays are from 2 to 100 times bigger than the measured delays and after 60 Mbps the communication between the client and the server is interrupted.

IEC 61850 service's time performances are the same in the two different architectures and Figure 71 shows the results. Time performances of the tested IEC 61850 services remain almost constant up to 40% of available bandwidth. Above 60%, the communication between the client and the servers crashes.

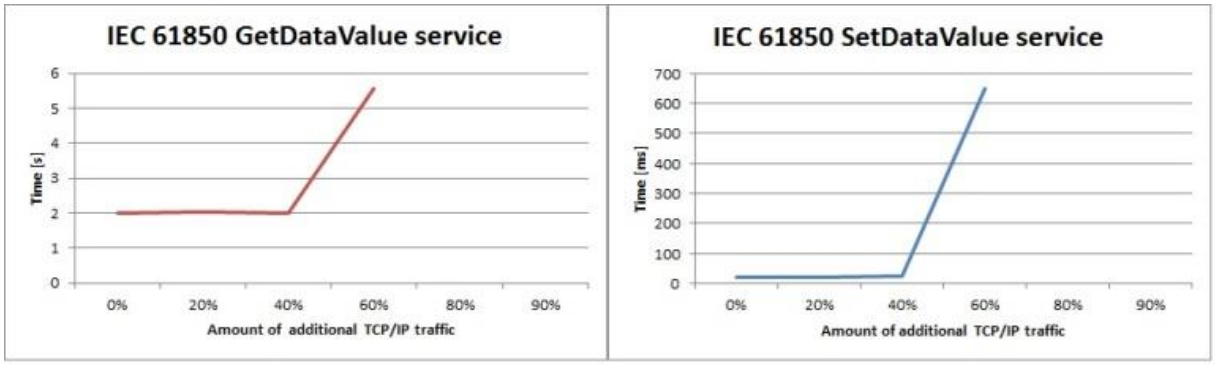


Figure 71 IEC 61850 network time performance

The results of the tests show that all the tested architectures present an approximated limit of usage of the bandwidth around 20%. Under this limit all the devices in the network runs without losing their time performances and without operating failures. Moreover, the limit of 20% of bandwidth usage allows the network to run some typical TCP/IP applications like video surveillance. A video streaming for surveillance can be overestimated with 1Mbps per camera: the available bandwidth allows installing a significant number of cameras on the network.

However, the failures of the I/O device with address 192.168.1.113 and the proper operation of the other I/O devices suggest that high performance devices are required to carry out time critical applications.

Comparing the time performances of the two architectures it seems that the network card of a common Personal Computer is more performing than the network card installed in a PLC. This fact together with the better and deeper integration that the Soft-PLC offers, are the main reasons to say that the PC-based architecture is recommended only for low-demand applications. For motion control or other high demand applications, specific hardware is required, and in these cases PLC based solutions are preferred.

2.5.6 State of the art solution: Time Sensitive Networking

Previous chapter highlights the problem of the available bandwidth needed by industrial automation communication to work without failures. The solution of this issue can be an improved determinism of the message exchange. Considering the

technologies used and descripted in this work, Profinet already provides hard real time functionalities using synchronization and network topology scheduling. This is called Profinet IRT, Isochronous Real Time, and it requires dedicated and specific hardware. A standard solution is Time Sensitive Networking. TSN is a set of standards developed by IEEE 802.1 working group as an extension of IEEE 802.1Q (VLANs) to make Ethernet standard deterministic, more robust and reliable. TSN is a layer 2 series of standard that is designed to be fully integrated in Ethernet stack. The three main features that TSN adds to an Ethernet network are:

- Time synchronization;
- Scheduling;
- Network configuration.

Time synchronization is mandatory for hard real time communication. All devices in the network need to have a common time reference and need to synchronize their clocks among each other. IEEE 802.1 working group ha developed a dedicated TSN time synchronization standard called IEEE 802.1AS, based on IEEE 1588v2 PTP profile. The working principle is a synchronization using peer delay. A device on the network is selected as Grand Master and synchronizes with a slave with a message exchange. The slave device then acts as a master to its slave and so on.

Scheduling allows and traffic shaping allows the co-existence of different kind of traffic, each with different needs in term of latency and bandwidth. IEEE 802.1Q already permits the use of 8 different traffic classes to prioritize frames but there is no guarantees on delivery time caused by the buffering effect of Ethernet switching. TSN introduces a new processing method such as IEEE 802.1Qbv time aware traffic scheduler. This scheduler is designed to separate the communication on the Ethernet network into fixed length, repeating time cycles. Within these cycles, different time slices can be configured that can be assigned to one or several of the eight Ethernet priorities. By doing this, it is possible to grant exclusive use - for a limited time - to the Ethernet transmission medium for those traffic classes that need transmission guarantees and can't be interrupted. By granting exclusive access to the transmission medium and devices to time-critical traffic classes, the buffering effects in the Ethernet

switch transmission buffers can be avoided and time-critical traffic can be transmitted without non-deterministic interruptions.

Other important aspect to reach deterministic Ethernet through TSN are fault-tolerance (IEEE 802.1CB) and configuration (IEEE 802.1 Qcc). The first is used to provide lightweight redundancy for reliable delivery of traffic streams. The latter define the management interfaces to enable TSN network administration.

3. SAFETY ANALYSIS

3.1 FUNCTIONAL SAFETY OVERVIEW

In this chapter, an overview of safety and functional safety concepts will be provided. In this regard, a quick survey of the corresponding IEC standard will be supplied, focusing on the concepts more useful for this thesis.

Instrumented safety systems are not new. It has long been the practice to fit protective systems to industrial process plant where there is a potential threat to life or to the environment. These systems are independent of the normal process control, and act to render the plant safe in the event of a malfunction.

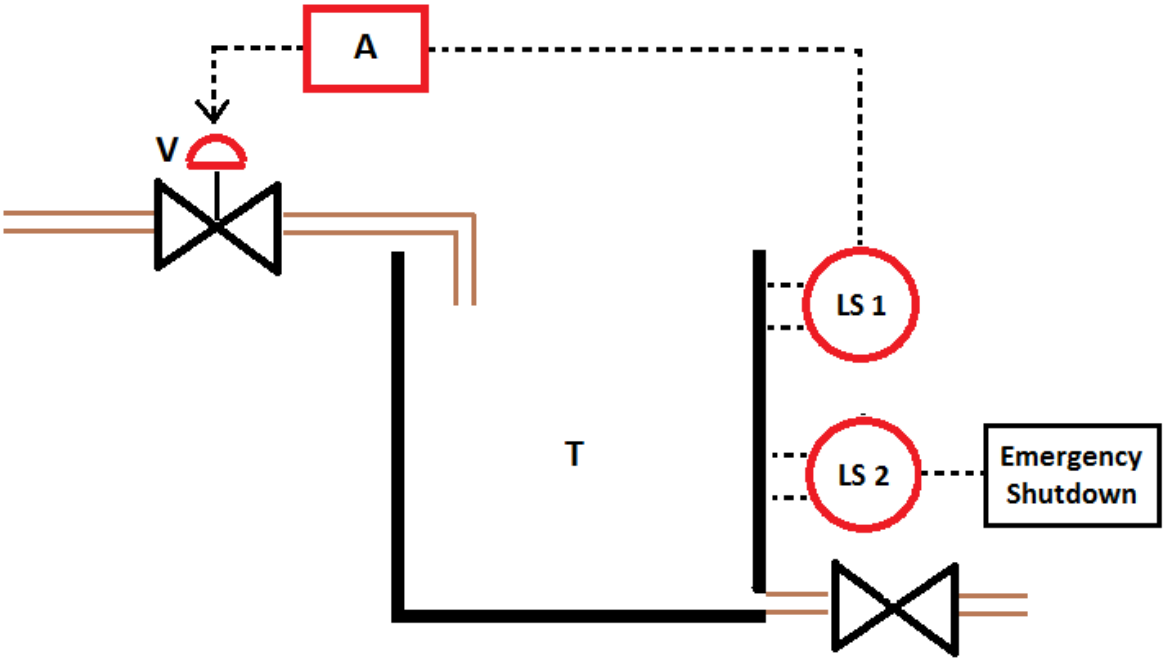


Figure 72 Example of a basic process

Supposing to have a system constitute by a tank (T) containing liquid used to cool down a reactor, two level switch (LS), an actuator (A) and a valve (V). The lack of the liquid inside the tank can lead to failure of the reactor cooling system and, therefore, a possible hazard for people and the environment.

This constitutes a Risk, which is combination of the probability of occurrence of harm and the severity of that harm. It is different from the concept of probability, which is the likelihood for a generic harmful event to occur, the risk is the likelihood of a generic hazardous event mixed with its consequences.

Who builds and manages a system must ensure that the risk introduced by the latter (e.g., reactor overheating) is below the level which is considered tolerable (risk which is acceptable in a given context, based on the current values of society). If this has not been the case, remedial actions must be considered, in order to reduce such a risk below the tolerable level. In this example, the risk can be reduced through an additional ESD (Emergency ShutDown) system that, in case of emergency, leads the system in a safe condition.

The overall program to ensure that the safety-related E/E/PE system (electrical/electronic/programmable electronic system) brings about a safe state when called upon to do so is defined as functional safety. Functional safety is the part of the overall safety that depends on a system or equipment operating correctly in response to its inputs. In other words, it is the detection of a potentially dangerous condition resulting in the activation of a protective or corrective device or mechanism to prevent hazardous events arising or providing mitigation to reduce the consequence of such a hazardous event. Functional safety relies always on active systems[13].

3.1.1 IEC 61508: Functional Safety of E/E/PE Safety-Related Systems

The IEC 61508 standard covers safety-related systems when one or more of such systems incorporates electrical/electronic/programmable electronic (E/E/PE) devices. These devices can include anything from solenoid valves, electrical relays and switches through to complex Programmable Logic Controllers (PLCs). The standard specifically covers possible hazards created when failures of the safety functions performed by E/E/PE safety-related systems occur.

This standard is not merely a technical guideline. Indeed, its primary subject is the management of safety, and it is within this context that it addresses the technical issues involved in the design and development of systems. The standard seeks to introduce

safety management and safety engineering, not only into software and system engineering, but also into the management of all aspects of systems. The basic philosophy behind the safety life cycle is to develop and document a safety plan, execute that plan, document its execution (to show that the plan has been met) and continue to follow that safety plan through to decommissioning. Changes along the way must similarly follow the pattern of planning, execution, validation, and documentation.

It consists of seven parts. The first four are “normative” (i.e. they are mandatory) and the fifth, sixth and seventh are informative (i.e. they provide added information and guidance on the use of the first four). Figure 73 shows such a structure of IEC 61508.

Part	Title
1	General Requirement
2	Requirements for E/E/PE Safety-Related Systems
3	Software Requirements
4	Definitions and Abbreviations
5	Examples of Methods for the Determination of Safety Integrity Levels (SIL)
6	Guidelines on the Application of Parts 2 and 3
7	Overview of Techniques and Measures

Figure 73 IEC61508 parts

The development of safety functions, which embody the main principles of the standard, requires the following steps:

- Identify and analyse the risks
- Determine the tolerability of each risk
- Determine the risk reduction necessary for each intolerable risk
- Specify the safety requirements for each risk reduction, including their safety integrity levels (SILs)
- Design safety functions to meet the safety requirements
- Implement the safety functions
- Validate the safety functions

Although the standard formally addresses only safety-related E/E/PE systems, it points out that safety functions may also be provided by other technologies (such as hydraulic

systems) or external facilities (for example, management procedures). The principles of the standard should be applied in all cases.

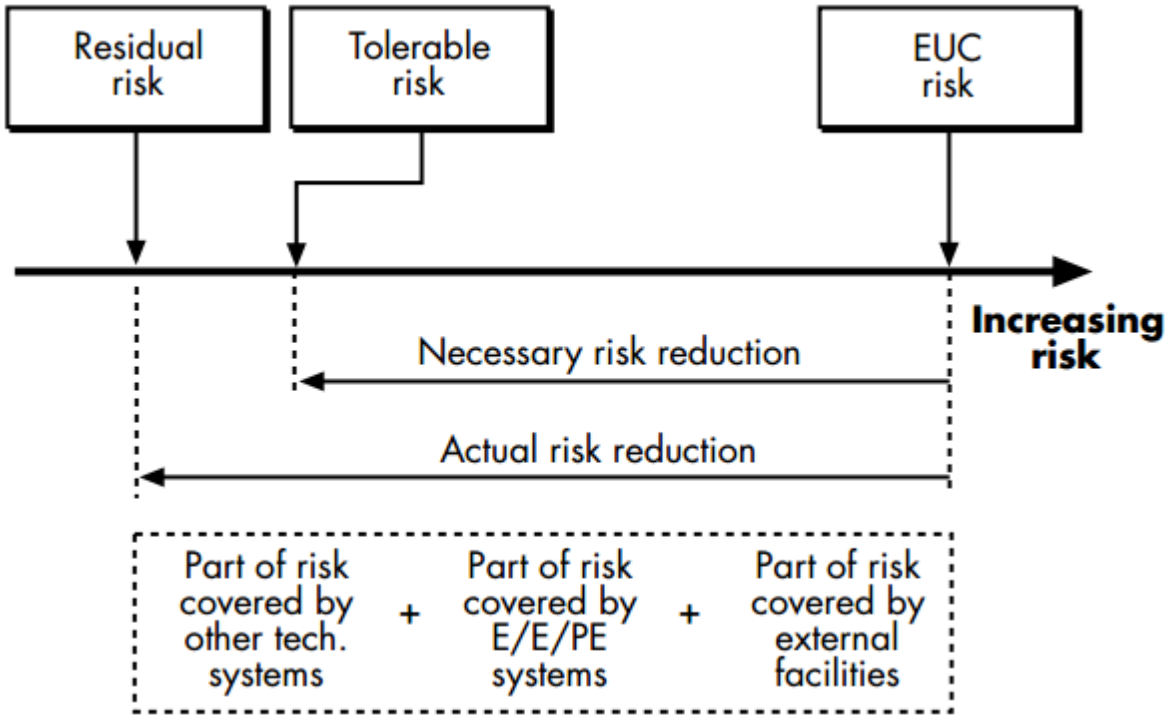


Figure 74 Combined risk reduction for the Equipment under Control EUC

Risk analysis is normally defined as consisting of three stages: hazard identification, hazard analysis, and risk assessment.

Hazard identification consists of an attempt to identify the potential sources of harm. For simple systems that have already been in operation for some time, methods such as brainstorming and the use of a checklist may be adequate. But for systems that are novel or complex, a team effort is required. A EUC (Equipment under Control) and its control system may pose many hazards, and as many as possible must be identified, for the risks associated with unidentified hazards will not be analysed or reduced. The importance of hazard identification cannot be emphasized too strongly, and the standard points out that identifying hazards concerned only with normal operation is not sufficient. Those arising from failures and 'reasonably foreseeable' misuse must also be identified.

Hazard analysis is the study of the chains of cause and effect between the identified hazards and the hazardous events (accidents) to which they might lead. The analysis is intended to determine causes and consequences, such that the risk attached to each hazard can be derived. It may be quantitative or qualitative. In a quantitative analysis, the probabilities of events are estimated, as are numeric values of their consequences. Then, the risks are calculated by multiplying the two. But qualitative analysis is also admissible, and the standard's definition of risk, as the combination of likelihood and consequence, facilitates this.

In the risk-assessment stage of risk analysis, the risk values determined in the previous stage are compared against tolerability criteria to determine if they are tolerable as they are and, if not, by how much they need to be reduced. There is necessarily a great deal of subjectivity in this process, not least in the decision of what level of risk is tolerable. It should be noted that tolerability may be different for each risk posed by the EUC and its control system, for it depends not only on the level of risk but also on the benefits to be gained by taking the risk and the cost of reducing it[14].

3.1.2 SILs and Probability of Failure

The working mode of each safety-related system can be classified in two categories, depending on the frequency of demands for operation:

- Low demand mode: where the frequency of demands for operation made on a safety-related system is no greater than twice the proof test frequency
- High demand or continuous mode: where the frequency of demands for operation made on a safety-related system is greater than twice the proof check frequency

Given a generic safety-related system working in low demand mode, it is possible to calculate and define its PFD (Probability of Failure on Demand), which is its probability of fail to perform its design function on demand. In other words, it is the safety unavailability of an E/E/PE safety-related system to perform the specified safety function when a demand occurs from the EUC or EUC control system. The average

probability of a system failing to respond to a demand in a specified time interval is referred as PFD_{avg}.

For systems working in high demand mode, it is possible to calculate and define their PFH (Probability Failure per Hour) which is comparable to a frequency, in fact, it is the average frequency of a dangerous failure of an E/E/PE safety related system to perform the specified safety function over a given period of time.

In order to standardize the risk reduction factor, IEC 61508 defines four lever of risk reduction called SILs (Safety integrity levels). In particular, they are order of magnitude levels of risk reduction. SIL1 has the lowest level of risk reduction, while SIL4 has the highest level of risk reduction.

Figure 75 shows the different SIL levels available and the corresponding PFD_{avg}.

Safety Integrity Level	PFD_{avg}	Risk Reduction Factor
SIL 4	$10^{-5} \leq PFD_{avg} < 10^{-4}$	$10^{4} < RRF \leq 10^{5}$
SIL 3	$10^{-4} \leq PFD_{avg} < 10^{-3}$	$10^{3} < RRF \leq 10^{4}$
SIL 2	$10^{-3} \leq PFD_{avg} < 10^{-2}$	$10^{2} < RRF \leq 10^{3}$
SIL 1	$10^{-2} \leq PFD_{avg} < 10^{-1}$	$10^{1} < RRF \leq 10^{2}$

Safety Integrity Level	PFH
SIL 4	$10^{-9} \leq PFD_{avg} < 10^{-8}$
SIL 3	$10^{-8} \leq PFD_{avg} < 10^{-7}$
SIL 2	$10^{-7} \leq PFD_{avg} < 10^{-6}$
SIL 1	$10^{-6} \leq PFD_{avg} < 10^{-5}$

Figure 75 Safety Integrity Levels

While the continuous mode appears to be far more stringent than the low demand mode, it should be remembered that the units for the continuous mode are per hour. The demand mode units assume a time interval of roughly one year per the definition. Since there are about 10,000 hours in a year (actual 8,760), the modes are approximately the same in terms of safety metrics.

Basically speaking, functional safety is achieved by properly designing a Safety Instrumented System (SIS) to carry out a Safety Instrumented Function (SIF) at a reliability indicated by the Safety Integrity Level (SIL).

Given a level of risk, theoretically it is always possible to further reduce it, with some benefits, at the cost of a certain technical and economic effort. No industrial activity is entirely free from risk and so many companies and regulators around the world require that safety risks are reduced to levels that are ALARP (As Low As Reasonably Practicable). The ALARP region lies between unacceptably high and negligible risk levels. Even if a level of risk for a "baseline case" has been judged to be in this ALARP region it is still necessary to consider introducing further risk reduction measures to drive the remaining, or "residual", risk downwards. The ALARP level is reached when the time, trouble and cost of further reduction measures become unreasonably disproportionate to the additional risk reduction obtained.

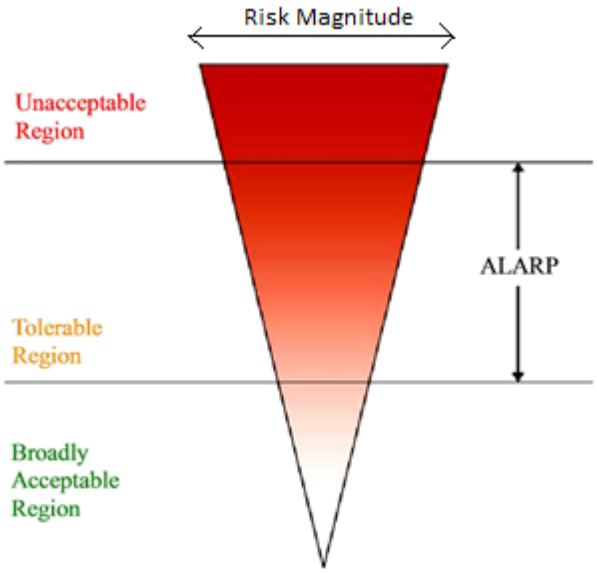


Figure 76 ALARP region

Although each system has its probability of failure, some of its failures can be considered safe and they are identified through the SFF (Safe Failure Fraction) indicator. The SFF is property of a safety related element that is defined by the sum of the rate of “safe” failures plus the rate of detected dangerous failures divided by the

whole failure probability. In case of constant failure rates, this ratio is represented by the following equation:

$$SFF = \frac{\lambda_S + \lambda_{DD}}{\lambda_T}$$

Where:

- λ_S is the probability of non-dangerous failure;
- λ_{DD} is the probability for a dangerous failure to be detected;
- λ_T is the whole failure probability.

Each system is as well characterized by an HFT (Hardware Fault Tolerance), which represent the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively designed system in which even a small failure can cause total breakdown. Basically, it represents the number of contemporary failures N, by which the (N+1)-th can result in the loss of a safety function.

IEC 61508 defines two categories (Type A and Type B) of safety-related subsystem, depending on their complexity: components that are described as simple devices with well-known failure modes and a solid history of operation belong to Type A, while devices that are complex components with potentially unknown failure modes (e.g., microprocessors, ASICs, etc.) belong to Type B.

Given the HFT and the SFF parameters, it is possible to define the limit on the safety integrity level achievable by any particular level of fault tolerant safety-related subsystem (Figure 77 and Figure 78).

Safe Failure Fraction	Hardware Fault Tolerance		
	N = 0	**N = 1**	**N = 2**
$X < 60\%$	SIL 1	SIL 2	SIL 3
$60\% \leq X < 90\%$	SIL 2	SIL 3	SIL 4
$90\% \leq X < 99\%$	SIL3	SIL4	SIL 4
$X \geq 99\%$	SIL 3	SIL 4	SIL 4

Figure 77 Type A safe failure fraction chart

Safe Failure Fraction	Hardware Fault Tolerance		
	N = 0	**N = 1**	**N = 2**
$X < 60\%$	Not Allowed	SIL 1	SIL 2
$60\% \leq X < 90\%$	SIL 1	SIL 2	SIL 3
$90\% \leq X < 99\%$	SIL 2	SIL 3	SIL 4
$X \geq 99\%$	SIL 3	SIL 4	SIL 4

Figure 78 Type B safe failure fraction chart

An example of determining the maximum SIL achievable is constituted by a valve (Type A) with the following parameters:

Low demand mode

- $\mathrm{PFD}_{\mathrm{avg}}=3*10^{-5}$
- HFT = 0 (no redundancy)
- SFF = 95%

With those parameters, the valve can be used in systems up to SIL 3; SIL 4 can't be reached without considering redundancy. It is worth mentioning that it is not correct to say such a valve can be certified for SIL 3, as the SIL definition applies to a safety function as a whole, not to the individual subsystems it is composed by.

3.1.3 Safety and communication networks

IEC 61508, as explained in the previous chapter, does not restrict the use of digital communication protocol for safety-related functions instead of the traditional wired system. In such case, as safety concepts apply both to the software and the hardware part, the communication system requires to be certified as well. In this respect, the normative reference is the IEC 61784-3: Functional safety fieldbuses – General rules and profile definitions[18].

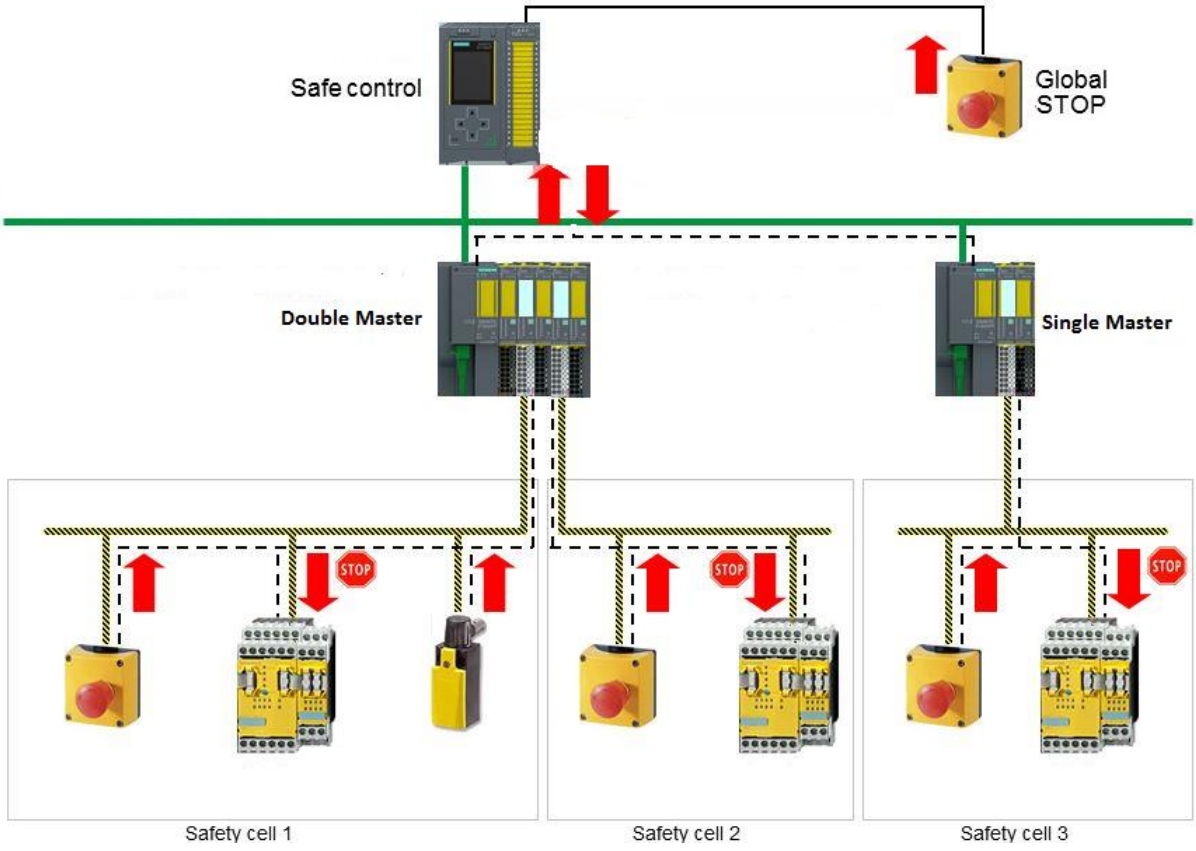


Figure 79 Example for communication network used for safety-related functions

Part three of IEC 61784 series explains some common principles than can be used in the transmission of safety-relevant messages among participants within a distributed network using fieldbus technology in accordance with the requirements of IEC 61508 series for functional safety. In the first part, it sets out all the features that communication systems used to carry out safety-related functions must have and the technical requirements they must operate under. In the second part, IEC 61784 specify

the certified communication profiles available nowadays to be used in safety related functions.

3.1.4 Black channel and White channel

Traditionally, if a signal needs to be sent from one controller to another, a hard wired output from one would be connected to the other and would be treated as any other safety input or output on the controller (Figure 80). This method is fine for a small number of signals but can become costly and difficult to modify as the number of signals increase. However, if the same data can be passed using some form of data link the system becomes more flexible and cost effective.

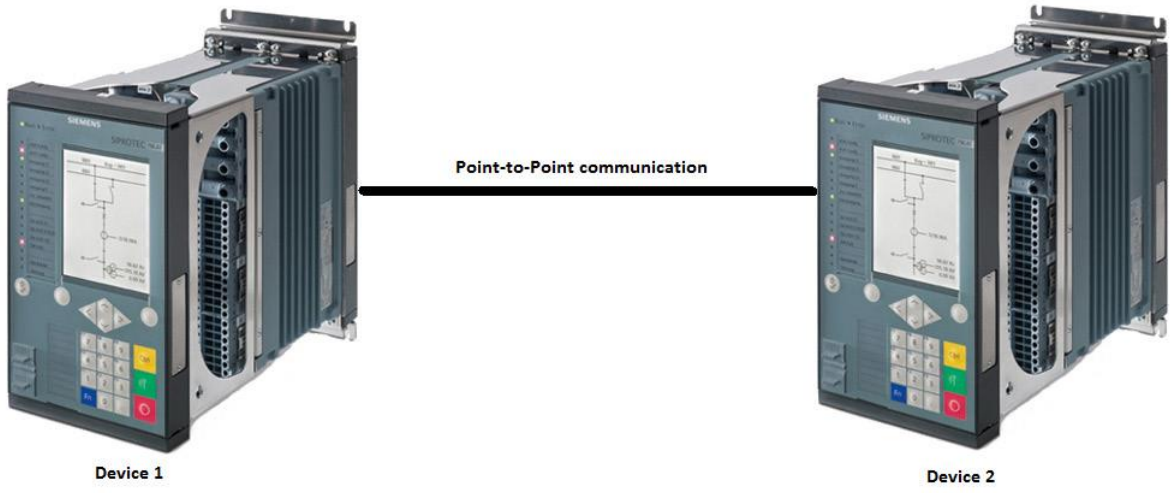


Figure 80 Point-to-Point link

Traditionally in these systems the data is handled by components that are known and considered part of the safety system; hence, their failure modes have been identified and appropriate measures taken to ensure that they meet the safety requirements. Any software installed on the communication components are also safety certified. Therefore, the communication channel is well defined and each configuration must be designed within the limits of the certified configurations. This type of communication channel is known as a White Channel as the properties of the channel are well defined and known.

Each of the components is designed with integrity levels that suit the specific application, such that there is confidence that the data is unlikely to be corrupted by,

for example, a software bug and any transmission errors are detectable. The downside of this design philosophy is that migrating to newer technologies can be slow and costly, as well as the components having inflexible architectures.

If we look at communication in general, the information technology arena has many powerful communication options that are both flexible and relatively cost-effective. One of the main drivers for this technology is obviously the high take up of the internet. For example, many personal computers can be connected at high speed using off-the-shelf low cost hubs and switches. The cost is driven down by the high volume of users (unlike the low number of users of white channel communication systems). The downside of using the high volume low cost unit is that its reliability can be an unknown quantity and the quality of the software inside is also an unknown factor; therefore, there is no way of knowing how a transported packet may be modified if a fault occurred.

Communication paths using such unknown components are called Black Channel communication paths.

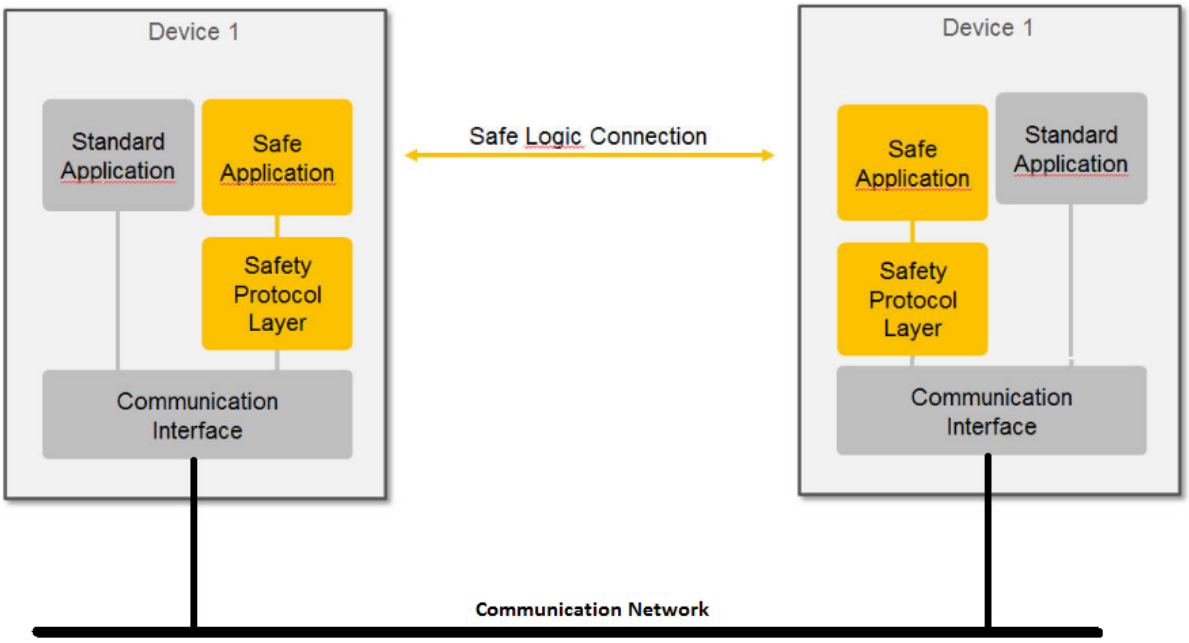


Figure 81 Example model of a function safety communication system

With a black channel, data from the sending safety system is launched into an unknown communication mechanism. Neither the sender nor receiver know the route the data might take, how many nodes will actually be handling the data, how long the data will take to get to the end and how many of the nodes may have interfered with the data before being received by the receiving safety system.

Therefore, to use a black channel for safety related data, the latter must have built-in mechanisms to detect any interference and with a confidence level of detection that is suitable for the safety application relying on the data. In order to do so, all the measures necessary to implement transmission of safety data in accordance with the requirements of IEC 61508 are performed by an additional Safety Communication Layer (Figure 81).

IEC 61508 uses the concept of the black channel or the white channel to define the requirements of the base fieldbus for transmission of safety data. Whether a communication channel is white or black is determined by where the safety measures are accomplished with respect to the base fieldbus. While IEC 61784 focuses on the use of IEC 61158 fieldbus based functional safety communication systems using the black channel approach, the following analysis aims to fully understand the requirements that a generic communication system must operate under when used to carry out safety functions[10].

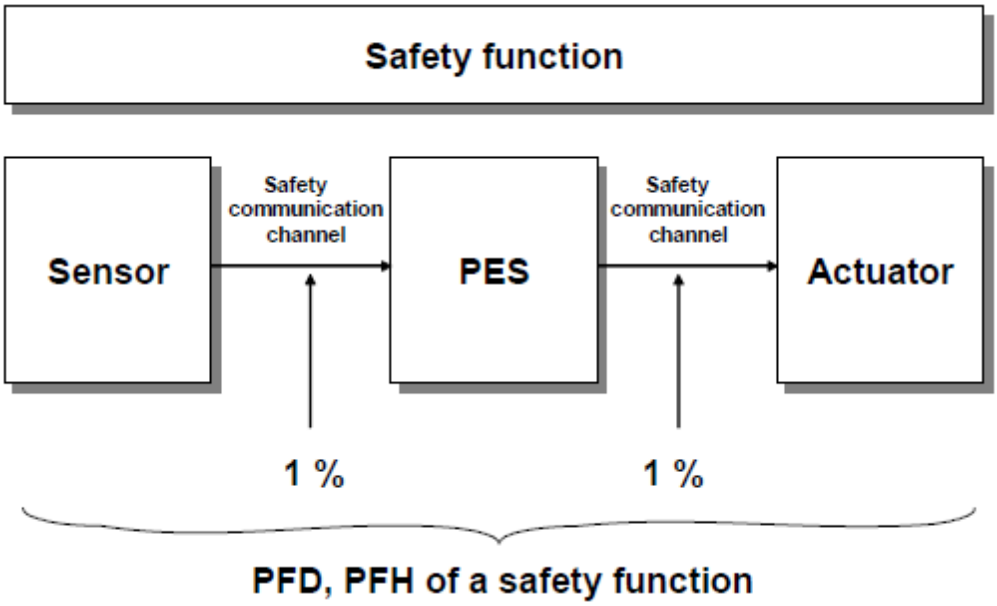


Figure 82 Safety communication as a part of Safety function

According to IEC 61508 a risk analysis will define the safety functions. These safety functions can be decomposed to parts that contribute to the overall safety function (for example sensors, safety communication channel and actuators). Regarding the communication system, as it provides transmission of safety data, the standard recommends it not to use more than 1 % of the maximum PFD or PFH of the target SIL which the functional safety communication profile is designed for (Figure 82).

3.1.5 Communication errors and Countermeasures

In order to use a communication system to interconnect devices carrying out safety-related functions, IEC 61784-3 defined eight types of communication error that may occur in such a communication system as well as some possible deterministic remedial measures.

Below is a list of the stated communication errors, a complete explanation of them will be provided in the next section:

- Corruption
- Unintended repetition
- Incorrect sequence

- Loss
- Unacceptable delay
- Insertion
- Masquerade
- Addressing

Below is a detailed view of the countermeasures, defined by IEC 61784, commonly used to detect deterministic errors and failures of a communication system, as contrasted to stochastic errors like message corruption due to electromagnetic interference:

- Sequence Number: A sequence number is integrated into messages exchanged between message source and message sink. It may be realized as an additional data field with a number that changes from one message to the next in a predetermined way.
- Time Stamp: In most cases the content of a message is only valid at a particular point in time. The time stamp may be a time, or time and date, included in a message by the sender. This indication implicitly requires the time base to be synchronized and, for safety applications, such a synchronization needs to be monitored.
- Time expectation: During the transmission of a message, the message sink checks whether the delay between two consecutively received messages exceeds a predetermined value. In this case, an error must be assumed.
- Connection authentication: Messages may have a unique source and/or destination identifier that describes the logical address of the safety relevant participant.
- Feedback message: The message sink returns a feedback message to the source to confirm reception of the original message.
- Data integrity assurance: The safety-related application process does not trust the data integrity assurance methods if they are not designed from the point of view of functional safety. Therefore, redundant data is included in a message to permit data corruptions to be detected by redundancy checks. If safety relevant

(SR) and non-safety relevant (NSR) data are transmitted via the same bus, different data integrity assurance systems or encoding principles may be used (different hash functions, for example different CRC generator polynomials and algorithms), to make sure that NSR messages cannot influence any safety function in an SR receiver.

- Redundancy with cross-checking: In safety-related fieldbus applications, the safety data may be sent twice, within one or two separate messages, using identical or different integrity measures, independent from the underlying fieldbus. In addition to this, the transmitted safety data is cross-checked for validity over the fieldbus or over a separate connection source/sink unit. If a difference is detected, an error must have taken place during the transmission, in the processing unit of the source or the processing unit of the sink. When redundant media are used, then common mode protection should be considered using suitable measures (for example diversity, time skewed transmission).

Figure 83 shows the correlation between the errors and the possible detection methods stated before.

Communication errors	Safety measures							
	Sequence number (see 5.4.2)	Time stamp (see 5.4.3)	Time expectation (see 5.4.4)	Connection authentication (see 5.4.5)	Feedback message (see 5.4.6)	Data integrity assurance (see 5.4.7)	Redundancy with cross checking (see 5.4.8)	Different data integrity assurance systems (see 5.4.9)
Corruption (see 5.3.2)					X [d]	X	Only for serial bus [c]	
Unintended repetition (see 5.3.3)	X	X					X	
Incorrect sequence (see 5.3.4)	X	X					X	
Loss (see 5.3.5)	X				X		X	
Unacceptable delay (see 5.3.6)		X	X [b]					
Insertion (see 5.3.7)	X			X [a]	X		X	
Masquerade (see 5.3.8)				X	X			X
Addressing (see 5.3.9)				X				

Figure 83 Overview of the effectiveness of the various measures on the possible errors

3.1.5.1 CORRUPTION

Data corruption refers to errors in data that occur during writing, reading, storage, transmission, or processing, which introduce unintended changes to the original data (Figure 84). In a communication system, messages may be corrupted due to errors within a bus participant, due to errors on the transmission medium, or due to message interference.

The possible countermeasures defined by IEC 61784 are:

- Feedback message
- Data integrity assurance
- Redundancy with cross-checking

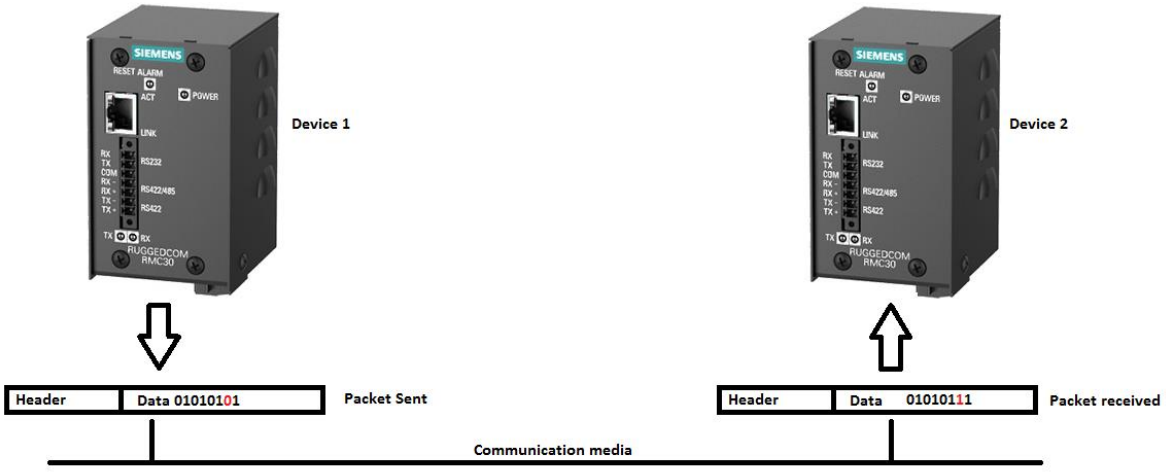


Figure 84 Example of Corruption

3.1.5.2 UNINTENDED REPETITION

Unintended repetition occurs each time, due to an error, fault or interference, old not updated messages are repeated at an incorrect point in time (Figure 85).

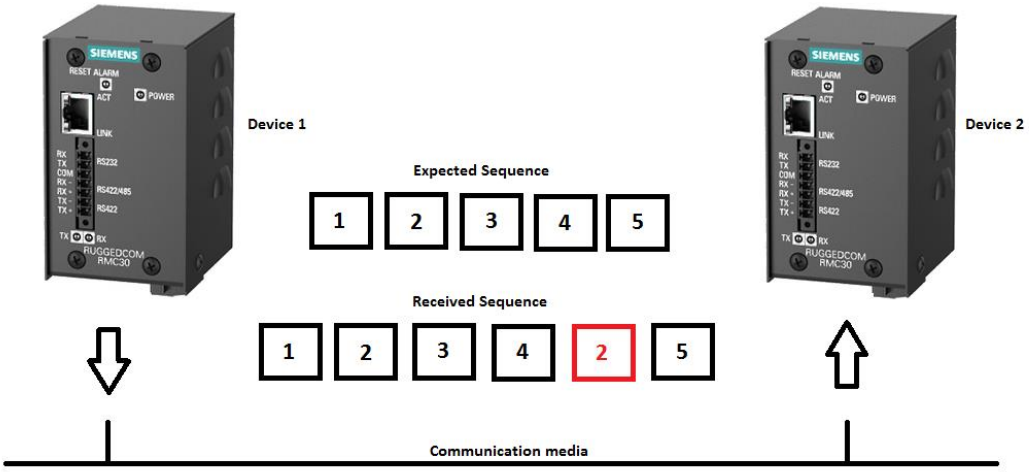


Figure 85 Example of Unintended repetition

Repetition by the sender is a normal procedure when an expected acknowledgment/response is not received from a target station, or when a receiver station detects a missing message and asks for it to be resent. In some cases, the lack of response can be detected and the message repeated with minimal delay and no loss of sequence, in other cases the repetition occurs later and arrives out of sequence with other messages.

The possible countermeasures defined by IEC 61784 are:

- Sequence Number
- Time Stamp
- Redundancy with cross-checking

3.1.5.3 INCORRECT SEQUENCE

Incorrect sequence occurs each time, due to an error, fault or interference, the predefined sequence (for example natural numbers, time references) associated with messages from a particular source is incorrect (Figure 86)

The possible countermeasures defined by IEC 61784 are:

- Sequence Number
- Time Stamp
- Redundancy with cross-checking

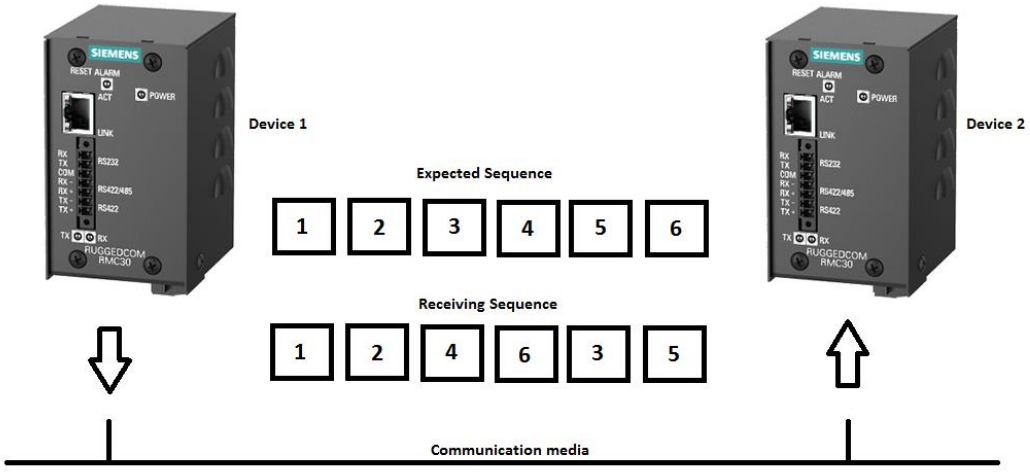


Figure 86 Example of Incorrect sequence

3.1.5.4 LOSS

Loss occurs each time, due to an error, fault or interference, a message is not received or not acknowledged (Figure 87).

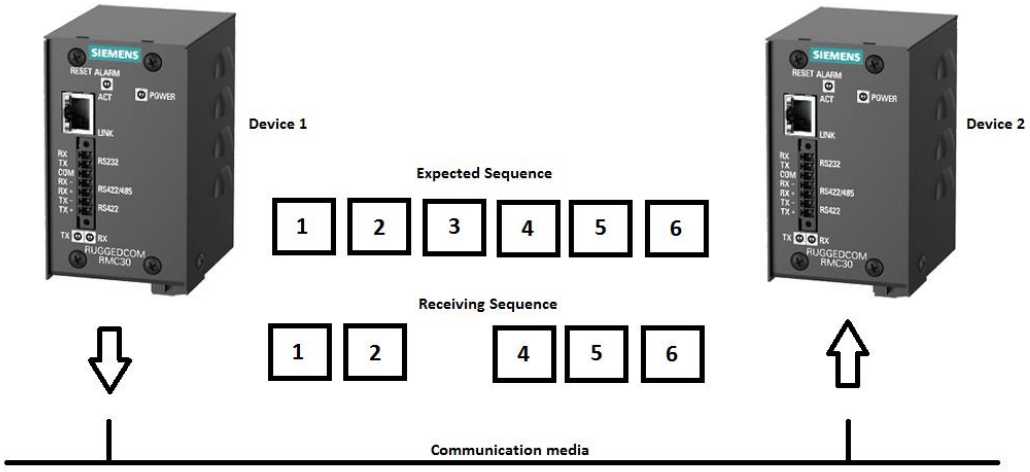


Figure 87 Example of Loss

The possible countermeasures defined by IEC 61784 are:

- Sequence Number
- Feedback message
- Redundancy with cross-checking

3.1.5.5 UNACCEPTABLE DELAY

Unacceptable delay occurs each time a message may be delayed beyond their permitted arrival time window. For example due to errors in the transmission medium, congested transmission lines, interference, or due to bus participants sending messages in such a manner that services are delayed or denied (for example FIFOs in switches, bridges, routers) (Figure 88).

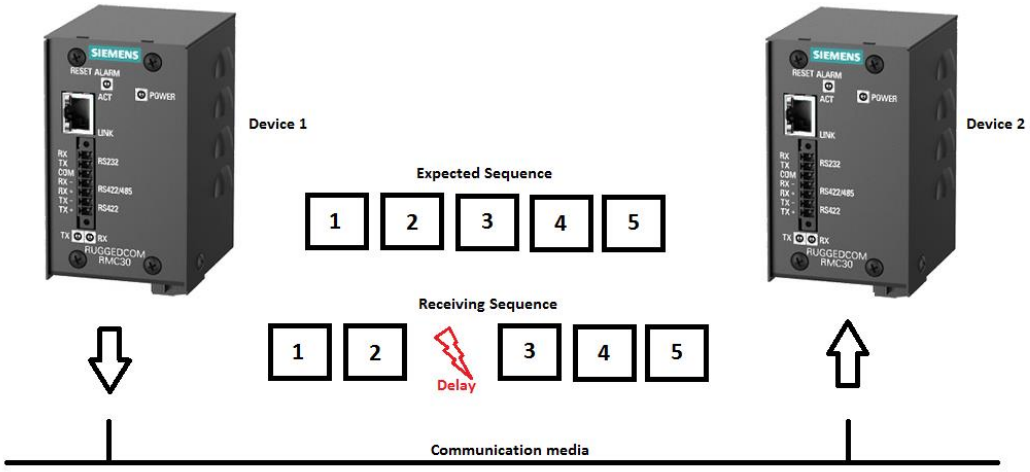


Figure 88 Example of Unacceptable delay

The possible countermeasures defined by IEC 61784 are:

- Time expectation
- Time Stamp

3.1.6 Insertion

Insertion occurs each time, due to a fault or interference, a message is inserted that relates to an unexpected or unknown source entity (Figure 89).

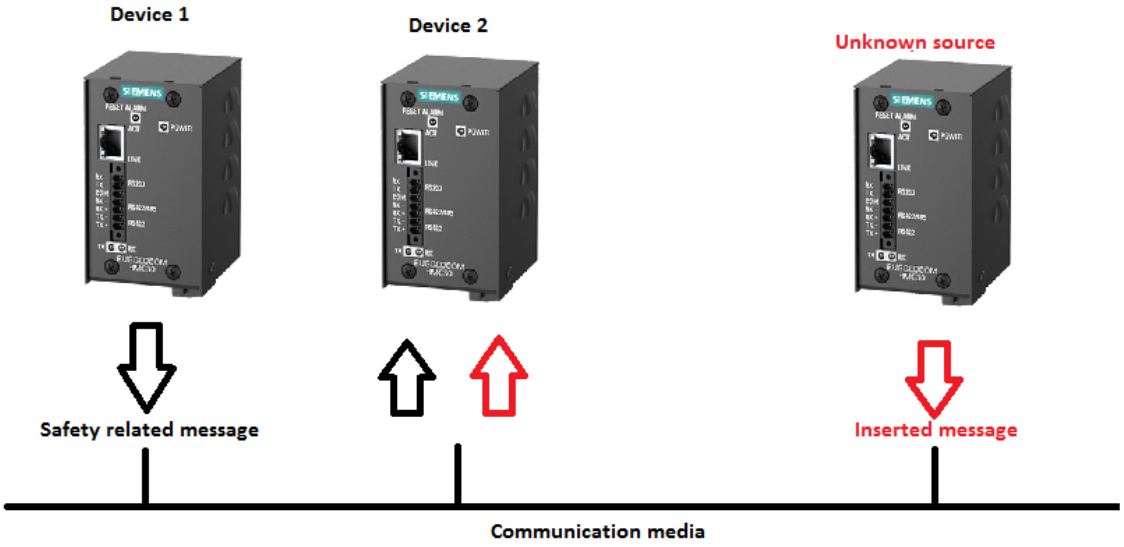


Figure 89 Example of Insertion

These messages are additional to the expected message stream, and because they do not have expected sources, they cannot be classified as Correct, Unintended repetition, or Incorrect sequence.

The possible countermeasures defined by IEC 61784 are:

- Sequence number
- Connection authentication
- Feedback message
- Redundancy with cross-checking

3.1.7 Masquerade

Masquerade occurs each time, due to a fault or interference, a message is inserted that relates to an apparently valid source entity, so a non-safety relevant message may be received by a safety relevant participant, which then treats it as safety relevant (Figure 90).

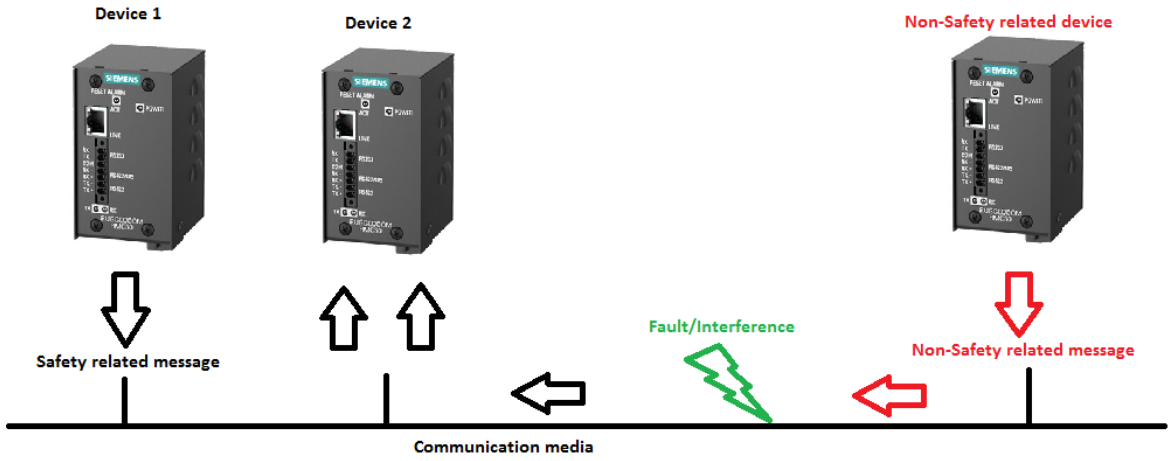


Figure 90 Example of Masquerade

The possible countermeasures defined by IEC 61784 are:

- Connection authentication
- Feedback message
- Different data integrity assurance system

3.1.8 Addressing

Addressing occurs each time, due to a fault or interference, a safety relevant message is sent to the wrong safety relevant participant, which then treats reception as correct (Figure 91).

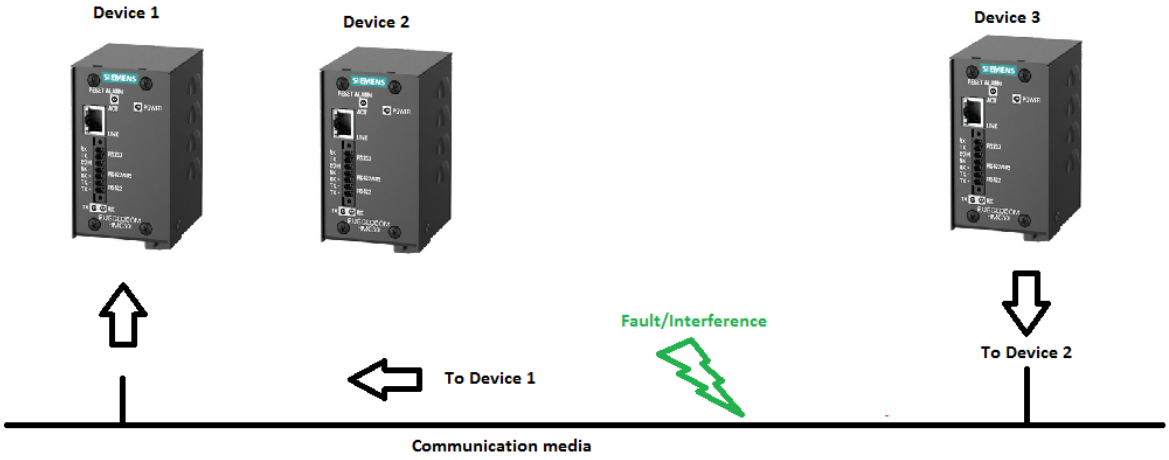


Figure 91 Example of Addressing

The only possible countermeasure defined by IEC 61784 is connection authentication.

3.1.9 Conclusions

During the previous sections, the communication errors that may occur in a communication system using black channel approach, as stated in IEC 61784, were analysed. For each of them, a list of the deterministic countermeasures that may be implemented in order to mitigate such a stochastic phenomenon was provided.

In the following chapters, a full overview of the IEC 61850 will be given, as well as a list of the possible communication mechanisms. In doing so, particularly attention will be given to the horizontal communication system stated in such a standard, also known as GOOSE.

Starting from the safety requirements for the communication network listed before, a comparison between them and the features provided by the GOOSE communication will be supplied. This analysis aims to demonstrate that the GOOSE communication

stated in IEC 61850 is compliant with the IEC 61784 series requirements and, therefore, it can be used for implementing safety-related functions.

3.2 IEC 61850 FOR SAFETY-RELATED FUNCTIONS

One of the goals of this thesis is showing that it is also possible to use the IEC 61850 standard to perform safety related functions.

As mentioned before, the relevant standard for functional safety operations provided through fieldbus communication systems is the IEC 61784-3. It identifies which features the communication system must fulfil to meet the standard's requirements. The fieldbus system, described by IEC 61784, uses the three ISO/OSI levels (physical, data link and application layer) by adopting a black channel approach to achieve the required safety functions. In regards of IEC 61850, and particularly for the GOOSE communication system, the three ISO/OSI levels are used and safety functionalities are implemented to guarantee a fast and reliable communication.

This chapter will highlight features and functions of GOOSE messages to determine whether, by their use, it is possible to render GOOSE communication compliant to IEC 61784-3 and allow them to be utilized for safety related application.

3.2.1 Communication Errors Using GOOSE Messages

As seen before, the GOOSE message contains some fields fully analogous to those requested by IEC 61784-3 for its compliance: Sequence Number, Time Stamp, Time Allowed to Live, and VLAN tagging.

Below is a list of possible communication errors defined in IEC 61784-3 and how they are manifested in a GOOSE communication:

- Corruption: GOOSE messages are encapsulated directly in an Ethernet packet allowing them to take advantage of this characteristic. A GOOSE message might experience corruption during its transmission from the sender to the receiver due to electromagnetic interferences, for its nature does not contain

special error correction mechanisms. The corrupted packet should be discarded by the receiving IED.

- Unintended repetition: it happens when a GOOSE message is repeated at an incorrect point of time. Every GOOSE message is characterized by a unique Sequence Number and State Number, such that if a repetition occurs, the receiver sees a packet containing a Sequence/State Number referring to an already acknowledged GOOSE, therefore it should not be considered. In this case, the Time Stamp refers to the point of time in which the state of the data set, which the GOOSE is linked to, has changed. This means that there might be messages with the same Time Stamp indication, for instance if the data set has not changed, but not with the same State and Sequence Number.
- Incorrect sequence: as described above, GOOSE messages contain a unique State and Sequence number indication always following an increasing integer sequence defined by the sending side. If, due to an error or an EMI, messages arrives in the wrong order, messages should not be acknowledged and the receiver should report such a deficiency. This happens because, depending on the communication layout, some Ethernet packets may use different paths to reach the same destination. Between these paths there are devices, such as switches for instance, that can alter the normal flux of information.
- Loss: if, due to an error or an EMI, some GOOSE messages are lost during the transmission process then the receiving device shall notice that the natural order of the messages is missing some State/Sequence Number. A GOOSE message is always repeated with its retransmission sequence, such that if a message is lost, the following message may reach its destination. Either way, it is still a symptom of a communication problem and an error must be reported.
- Unacceptable delay: each N-th GOOSE has a time window in which it can be subscribed as a valid GOOSE from the receiver. The allotted time window is defined by the TAL parameter contained in the (N-1)-th message. GOOSE transmission uses Ethernet which is non-deterministic for its nature. With Ethernet, every message may use different paths to reach the same destination. Additionally, each time a source tries to transmit something, it must wait until the media is clear, giving different transmission times for each GOOSE

message. If a message is delayed beyond its time window, it should not be classified as valid and an error must be reported.

- Insertion: as the Ethernet is a shared media, it usually contains both safety and non-safety related traffic so that an insertion of traffic by an unknown source might be expected. Usually IEDs send GOOSE messages in multicast mode and only those IEDs programmed to subscribe to those GOOSEs are supposed to acknowledge them. Therefore, an IED is supposed to acknowledge only those GOOSE messages containing the right GoID, gocbRef and dataSet fields so that all the other messages, even if received, are ignored.
- Masquerade: as described above, the Ethernet is a shared media so that it contains both safety related and non-safety related messages. Every Ethernet device is distinguished by its unique MAC address such that, due to a fault or an interference, this MAC address can experience an alteration to match the one of a safety related source, thus treated as a safety relevant message by the receiver. As IEDs from many manufacturers do not check the source MAC to say whether a message can be classified as valid or not, a way to separate safety relevant messages from the non-safety relevant ones must be implemented.
- Addressing: it has already been said that usually GOOSE messages work with multicast mechanism so that there is not a real receiver defined by the sender side; receivers of a generic GOOSE are just the ones which are programmed to subscribe to that specific GOOSE.

3.2.2 Solutions available on the market

Nowadays, the market has matured in its use of fieldbus solutions for functional safety. In fact, it is relatively easy to find devices which implement at least one of the communication protocols defined in IEC 61784 for functional safety, particularly for Foundation Fieldbus, Profibus, Profinet and EtherCat solutions.

Regarding the IEC 61850 solutions, the matter is slightly different because most of the manufacturers provide this communication protocol for some of their devices but,

despite its first edition released in 2007, none of those has implemented a secure GOOSE communication compliant to IEC 62351-6 yet.

This analysis starts looking at the available features provided by three main manufacturers of IEC 61850 IEDs: ABB, SEL and Siemens. This analysis uses these features to implement the functions, focusing on the GOOSE communications and the IEC 61784-3 requirements.

3.2.2.1 SIEMENS SOLUTION

To make better use of the possibilities of this device and to facilitate engineering operations, Siemens supplies its devices with a software called DIGSI which, depending on the IED's features, provides accessibility to all the device functions and facilitates programming operations.

For both the analysis part and the implementation part, SIPROTEC 4 devices have been considered. SIPROTEC 4 leads the way in integrating protection, control, measurement, and automation functions in one device. In many fields of application, all secondary technical functions can be carried out using just one device. This lowers investment costs, reduces installation work in every respect, and increases system availability.

Upon receipt of a GOOSE message, IEDs whose configuration impose the subscription of that GOOSE check the following parameters to declare whether a message is valid or not:

- Destination MAC address
- EtherType owned by the Ethernet packet
- APPID
- gocbRef
- dataSet
- GoID
- State Number
- Sequence Number

- Test/Simulation bit
- confRev
- ndsCom
- numDatSetEntries

As can be seen, a check of the parameters' State Number and Sequence Number is carried out immediately. Indirectly, TAL is used to verify whether the message has arrived in the right time window or communication problems have been experienced.

In case of a non-valid message, due to any one of the parameters listed above, the message is ignored by the IED which looks forward to receive a valid message until the TAL expires, before reporting a communication problem.

Through DIGSI, it is possible to implement appropriate indicators on the validity of each received message by means of DIGSI CFC (Continuous Function Chart), which provides a graphic interface allowing to create logical functions between internal and external device parameters.

Once GOOSE communication has been configured, it is possible with the DIGSI CFC to configure each GOOSE subscriber to pass the received information through a logical block called SI_GET_STATUS (Figure 92).

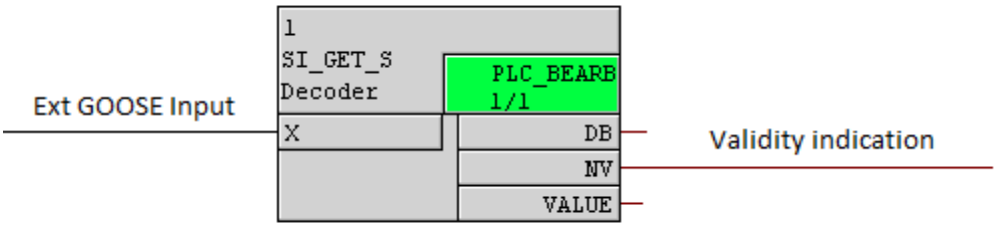


Figure 92 SI_GET_STATUS block

This logical block takes as input the Single Point indication sent by the GOOSE, it checks its validity and, among its output, there is one binary output (NV) indicating whether the message is classified as valid (0) or not (1).

Every function which you program with DIGSI CFC must be assigned to a priority class. The individual priority classes differ in their processing priority and execution time:

- Fast PLC processing (PLC_BEARB / PLC)
- Slow PLC processing (PLC1_BEARB / PLC1)
- Measured value processing (MW_BEARB/ MEASURE)
- Interlocking (SFS_BEARB / INTERLOCK)

To implement such a function, the PLC_BEARB class is used since Boolean inputs are not a trigger event in the MW_BEARB (measured value processing layer) and SFS_BEARB (interlock processing layer) and may therefore remain undetected if the event or signal is shorter than the execution interval of the particular processing layer. Particularly, functions in PLC_BEARB priority class are processed event-controlled with the highest priority: every change to a logical input signal is processed immediately.

In order to detect a GOOSE communication error, the behaviour of SIPROTEC 4 devices, when the received GOOSE message contains a different parameter from the expectation, is listed below:

- If a valid message is not received in the allotted time window equal to 2*TAL, the validity indication changes from 0 to 1;
- If a message does not arrive with the expected Sequence Number, given an expected Sequence Number N-th, only those messages containing such a parameter equal to N-th and (N+1)-th are acknowledged, all the others are ignored;
- If a message does not arrive with the expected State Number, given an expected State Number N-th, only those messages containing such a parameter equal to N-th and (N+1)-th are acknowledged, all the others are ignored;
- All duplicated messages are ignored;
- If a message arrives with an inconsistent gocbRef, GoID, and dataSet fields, the message is classified as not valid and an error is reported;

- All the incoming message, where any of the other parameters have been altered, are ignored.

Basically, the IED tolerates a missing telegram if the next telegram is received within its time allowed to live time out detection (the time allowed to live timeout detection occurs after 2 * TAL).

In the presence of communication errors, some give an immediate indication of invalidity, the remaining messages give such an indication after the time windows is elapsed.

Unfortunately, the SIPROTEC 4 Siemens family devices does not make any of the GOOSE parameters directly available to create logical functions (for instance Sequence Number, State Number, Time Allowed to Live and Time Stamp) but they do provide an indirect indication through the Status indication associated to a message once it is received.

3.2.2.2 ABB SOLUTION

ABB allows the download of its programming software directly from its website. The software is called PCM600 and it can be used to program all devices carrying an IEC 61850 interface.

The range of IEC 61850 available devices appears to be wide but with similar features in terms of GOOSE communication. In order to analyse their characteristics, devices of the 615 series has been used as reference.

ABB 615 series IEDs have been designed around IEC 61850. This means that the functionality of the IED is represented in a data model in accordance with the standard and the IEDs support a range of services provided by the standard.

REF615 is a dedicated feeder protection and control relay for protection, control, measurement and supervision of overhead lines and cable feeders in utility and industrial power distribution systems, including radial, looped and meshed distribution networks, with or without distributed power generation.

PCM600 uses IEC 61850 over the Ethernet to communicate with bay IEDs. This communication allows PCM600 to configure and monitor the IEDs. In addition to IEC 61850, the IEDs have optional communication protocols and hardware to connect to station engineering tools. PCM600 provides the ability to export the configuration of the IEDs or entire substation in a standard file format which allows for station engineering. The dedicated software PCM600 allows to configure all the aspects of the IED through internal logics as well as through all the IEC 61850 communication parameters.

In order to detect possible communication errors in a GOOSE transmission, ABB provides two groups of indicators, each of them regarding a specific kind of problem; both of the families are accessible by the LN LD0.GSEGGIO1 and the available information are showed in Figure 93.

REF6152 - IED Users | REF6152 - Application Configuration | **REF6152 - Parameter Setting**

Group / Parameter Name	IED Value	PC Value	Unit	Min	Max
GSELPRT1: 1					
GSE					
Outputs					
ALARM		False			
Monitoring					
Reset counters		False			
Received msgs		0		0	10000000
Transmitted msgs		0		0	10000000
State changes		0		0	10000000
SeqNum changes		0		0	10000000
Test msgs		0		0	10000000
State warnings		0		0	10000000
Seq. warnings		0		0	10000000
Recv. timeouts		0		0	10000000
ConfRev errors		0		0	10000000
NdsComm errors		0		0	10000000
Dataset errors		0		0	10000000

Figure 93 ABB IEC61850 communication alarms and warnings

The first signal considers the alarm trigger (red square in Figure 93) called GSEGGIO ALARM. It is available by means of the Signal Matrix (where all the input signals are

made available to create logical functions), which switches its value to true (1) every time one of the following events occurs:

- The time window (2*TAL) expires without reception of a valid message;
- A message arrives with the ndsCom bit value equal to 1;
- A message arrives with the confRev bit value not expected;
- A corrupted message arrives.

In the case where the incoming message creates the conditions for an alarm indication, the message is not processed and the device looks for a valid message within the time window.

The second family of signals is defined as Monitoring indicators (yellow square in Figure 93), which are not accessible to create logical functions. They include indicators on the correspondence of some GOOSE parameters to the expected ones and provides a clear indication of the reason for the alarm (if it has been triggered).

Unless otherwise stated, the messages which cause this kind of indication are acknowledged and therefore they are classified as valid. Practically this second group of indicators constitutes a sort of counter and they are not available as data to create a logical function.

Among other indicators, it is worth mentioning the presence of:

- Test msgs: this indicator is incremented every time a GOOSE message with the test bit equal to 1 is received.
- State warnings: this indicator is incremented every time a GOOSE message with a State Number different from the expected one arrives.
- Seq. warnings: this indicator is incremented every time a GOOSE message with a Sequence Number different from the expected one arrives.
- Recv. Timeouts: this indicator is incremented every time a valid GOOSE message is not received within the allowed time window; this error triggers also the alarm.

- ConfRev errors: this indicator is incremented every time a GOOSE message with a ConfRev field different from the expected one is received; this error triggers also the alarm and the message is not acknowledged.
- NdsComm errors: this indicator is incremented every time a GOOSE message with ndsCom bit equal to 1 is received; this error triggers also the alarm and the message is not acknowledged.
- Data errors: this indicator is incremented every time a GOOSE message with corrupted data is received; this error triggers also the alarm and the message is not acknowledged.

In conclusion, the only acknowledged messages are those whose parameters match the expected ones and those with altered State Number and/or Sequence Number (in the latter, the corresponding indication is incremented but the alarm is not triggered).

All the other messages create a direct alarm indication or they are ignored leading to a timeout error (TAL). The following Figure 94 summarize the action performed by the IED upon reception of a GOOSE.

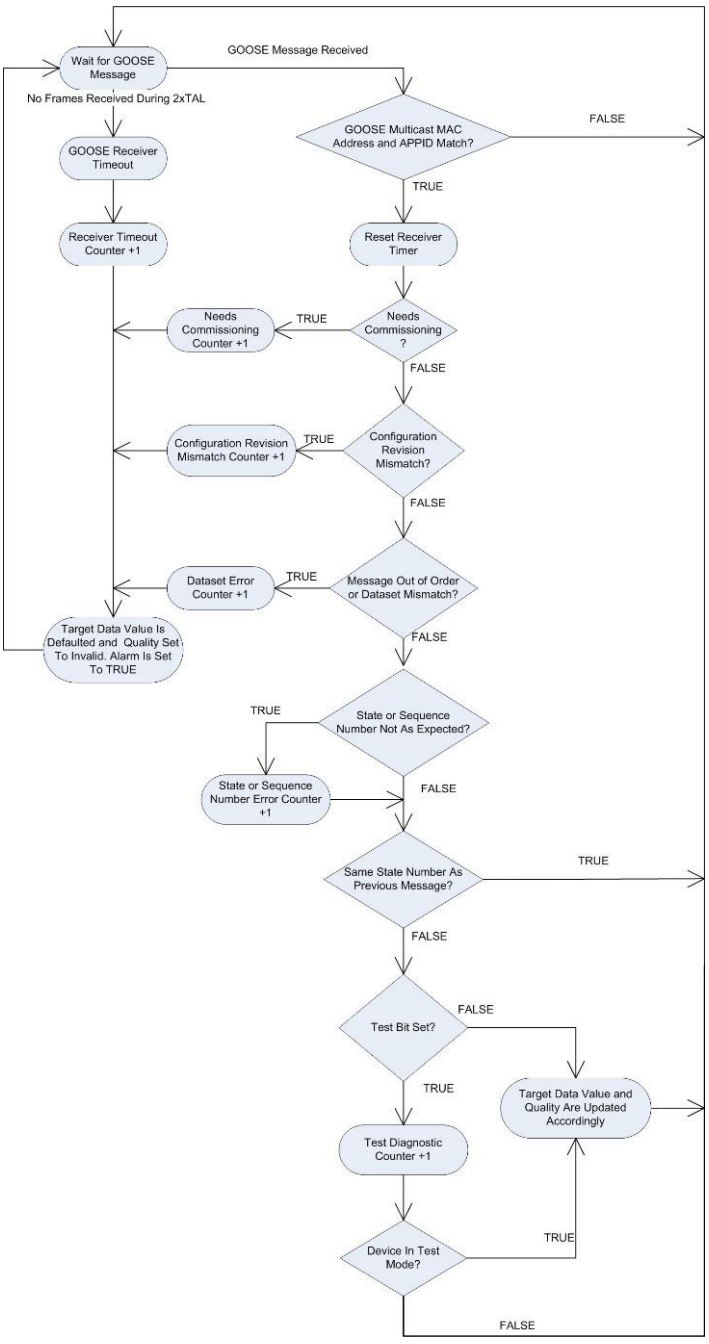


Figure 94 Receiving GOOSE data in 615 series IEDs

3.2.2.3 SEL SOLUTION

SEL (Schweitzer Engineering Laboratories) is an American company which has manufactured products in the United States since 1984 and now serves customers worldwide.

SEL supplies its IEC 61850 devices with a set of specific engineering software called acSELerator, each of which allows to access to a specific IED's feature. In this regards, acSELerator QuickSet is a tool for engineers and technicians to quickly and easily configure, commission, and manage devices for power system protection, control, metering, and monitoring. AcSELerator ARCHITECT is specialized in the substation communication, particularly it allows to manage both IEC 61850 MMS and GOOSE protocols. AcSELerator DIAGRAM BUILDER is a graphical, easy-to-use tool that helps users quickly and easily configure the SEL Real-Time Automation Controllers (RTACs).

The range of IEC 61850 available devices appears to be wide but with similar features in terms of GOOSE communication. In order to analyse their characteristics, devices of the SEL-400 group has been used as reference. SEL-400 devices provide a Protection, Automation, and Control System for high-speed distance and directional protection and complete control of a two-breaker bay.

In order to implement a fail-safe GOOSE communication compliant with IEC 61784-3, the GOOSE communication needs to be configured through the acSELerator ARCHITECT software mentioned before.

Figure 95 shows the GOOSE configuration where both Sender and Receiver IEDs were configured (brown square). The Sender was configured to publish a GOOSE message linked to the state of one of its binary inputs (blue square). Among the available parameters, VLAN Tags can be configured (yellow square) as well as the retransmission mechanism and the LN such a message is linked to (red square).

The Receiver's configuration is depicted in Figure 96, where the Receiver is configured to subscribe to the GOOSE published by the Sender (blue and yellow squares). During the setup, the software provides the access to some GOOSE's quality parameters,

allowing to choose which one the subscriber should consider (red square). Such parameters play an important role in order to detect whether a message is valid or not, providing an effective mechanism to detect and report communication errors.

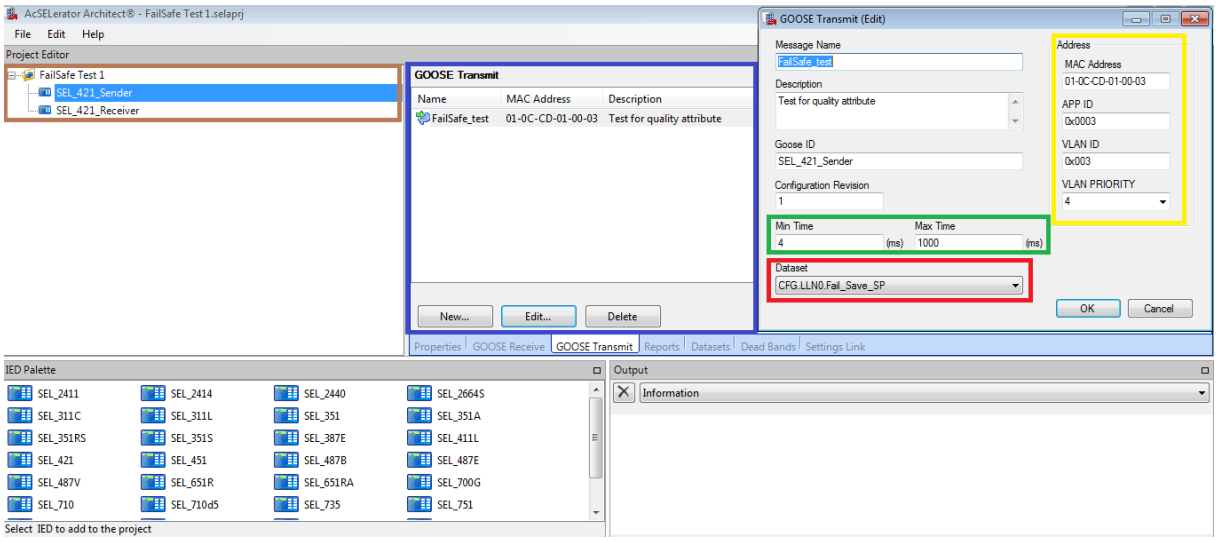


Figure 95 AcSELerator ARCHITECT – Sender's GOOSE configuration

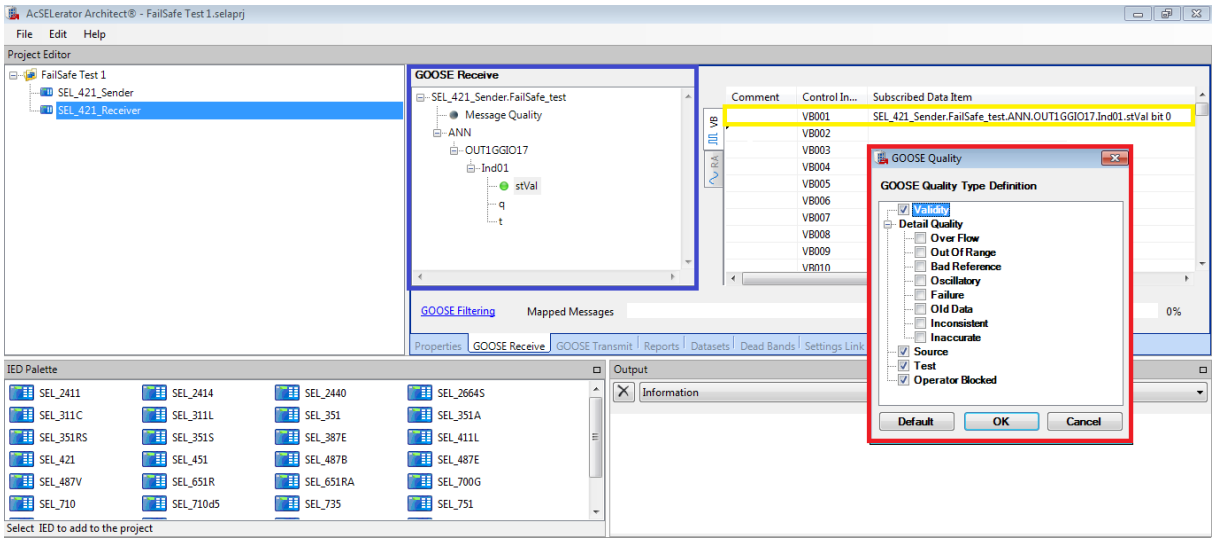


Figure 96 AcSELerator ARCHITECT – Receiver's GOOSE configuration

Upon receipt of a GOOSE message, IEDs whose configuration impose the subscription of that GOOSE check the following parameters to declare whether a message is valid or not:

- Destination MAC address
- EtherType

- gocbRef
- goID
- dataSet
- State Number
- Sequence Number
- Test bit
- confRev
- ndsCom
- numDatSetEntries

Each time a received GOOSE mismatches one of the previous parameters, either an indication is supplied and the message processed or the telegram is discarded without further indications. Such an indication is stored in a dedicated LN and can then easily be used in order to take the adequate countermeasures. In particular, below is a list of behaviours seemed interesting for the purpose of this thesis:

- If a message does not arrive prior the allotted time window, it is marked as lost and the corresponding error is issued.
- If GOOSE messages are skipped for any reason, the GOOSE subscriber will issue an error message and wait for the next message. For syntax errors, the GOOSE subscriber sets the appropriate error and waits for the next message.
- If a GOOSE is received out-of-order, the GOOSE subscriber sends an out-of-sequence error, and processes the received GOOSE message as normal.
- If a duplicated GOOSE is received, the GOOSE subscriber sends an out of sequence error, and processes the received GOOSE message as normal.
- All the other messages carrying a different parameter from the expectation are discarded without further indications.

In conclusion, some kind of communication failures lead to a direct error indication, while the others are issued through a time-out signal. Both ways a communication error is detected and reported. This quick overview has shown that all the three most important IEC 61850 device's manufacturers have implemented an effective way to address such an issue in their devices.

3.3 TESTING THE IEC 61850 STANDARD FOR SAFETY APPLICATIONS

This section aims to provide an actual implementation of the IEC 61850 for safety-related functions using the Siemens SIPROTEC devices described in 3.2.2.1.

Figure 97 Theoretical configuration

The initial configuration includes two IEDs exchanging GOOSE messages, one PC configured to detect all the traffic exchanged between IEDs and one managed switch compliant with IEEE 802.1Q (Figure 97).

The first protection is configured to send GOOSE messages associated with the logical state of its binary input (e.g., sending a GOOSE every time one input changes its logical state). The second one is programmed to receive and subscribe to the same GOOSE messages sent by the first one.

For this test, SIPROTEC 7SJ62 devices belonging to Siemens SIPROTEC 4 group is used.

The IED's network is configured as shown in Figure 98, where both of SIPROTEC devices are visible and one IEC 61850 Station which manages GOOSE communications.

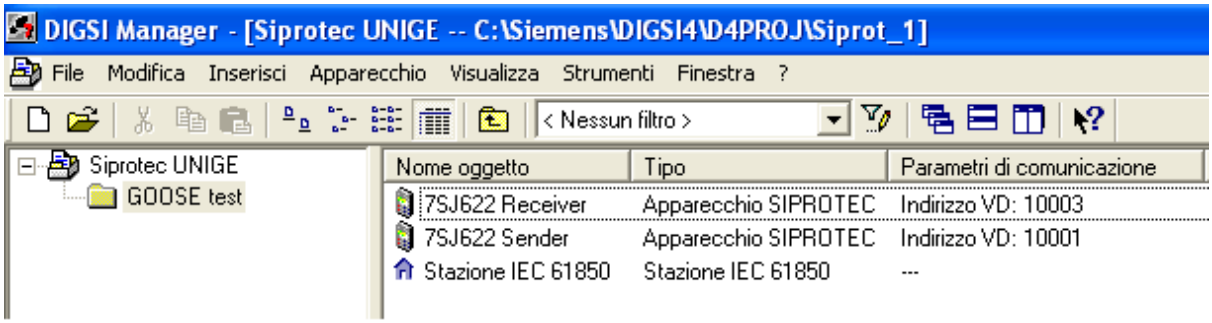


Figure 98 Digsi configuration

The Sender protection is setup to send a GOOSE message every time the function button F4 (front panel) is pressed. In order to do so, one Internal Single Point variable is defined in DIGSI CFC. It takes as input the function button F4 (yellow circle in Figure 99), then the signal is sent to the System Interface (brown circle in Figure 99) which handles GOOSE communications.

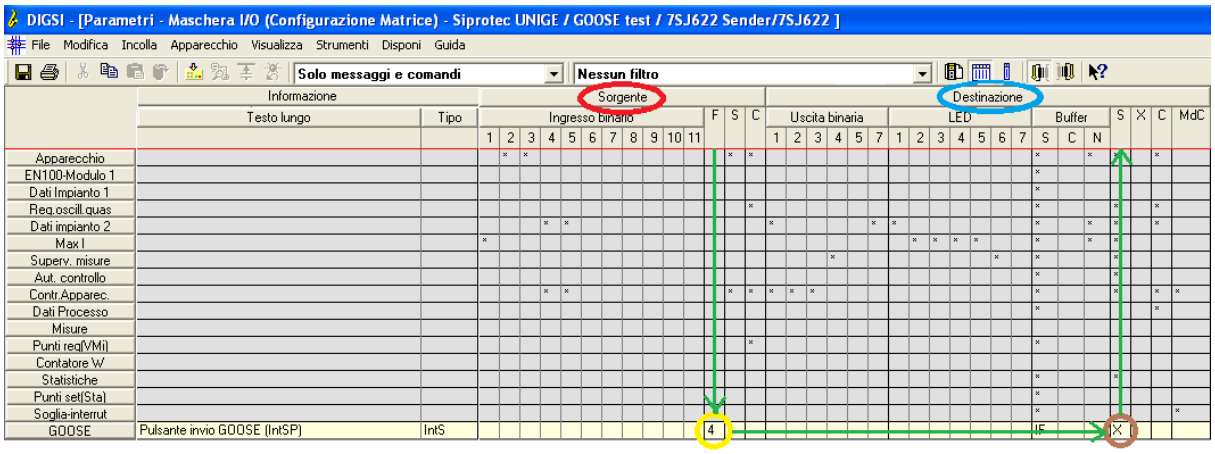


Figure 99 Sender I/O mask

The receiver is configured to subscribe exactly that particular type of GOOSE through system interface, then that signal is processed by a specific logic function to check its validity.

The receiving side configuration is shown in Figure 100; each single received GOOSE message with its corresponding Single Point indication with Status (SPS) parameter is read by system interface (circled in orange), then is forwarded to DIGSI CFC to create an appropriate internal logic (circled in yellow).

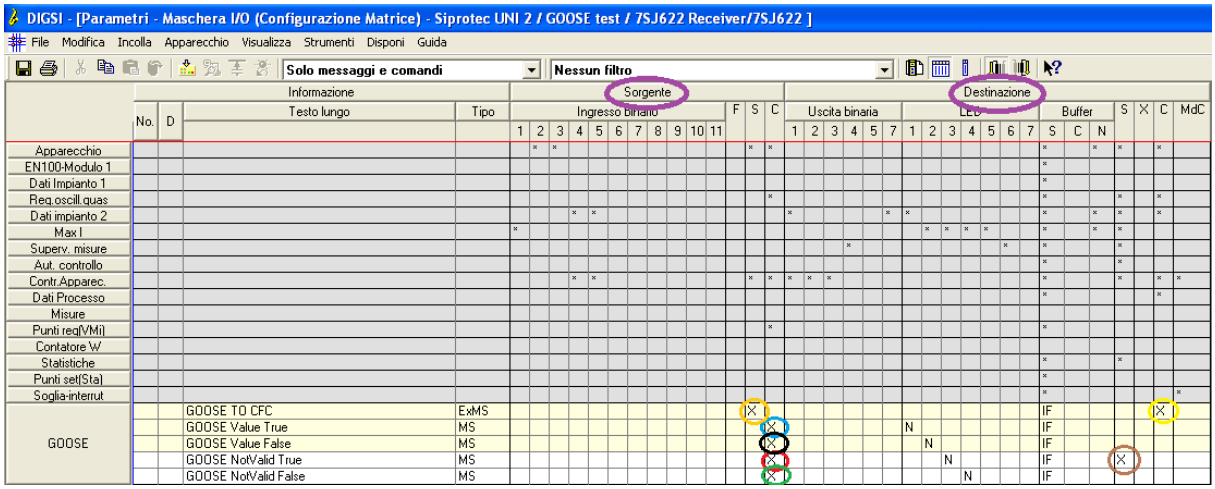


Figure 100 Receiver I/O mask

The internal logic implemented in such a device is shown in Figure 101. The received message is processed by SI_GET_STATUS logical block which, among its outputs, provides the current Binary Value associated to the SPS and its validity indication (to reveal possible communication errors).

As output, four LED indications are supplied, indicating the current value (true or false) and its status (valid or not valid); such information is shown in circles light blue, black, red and green in Figure 100.

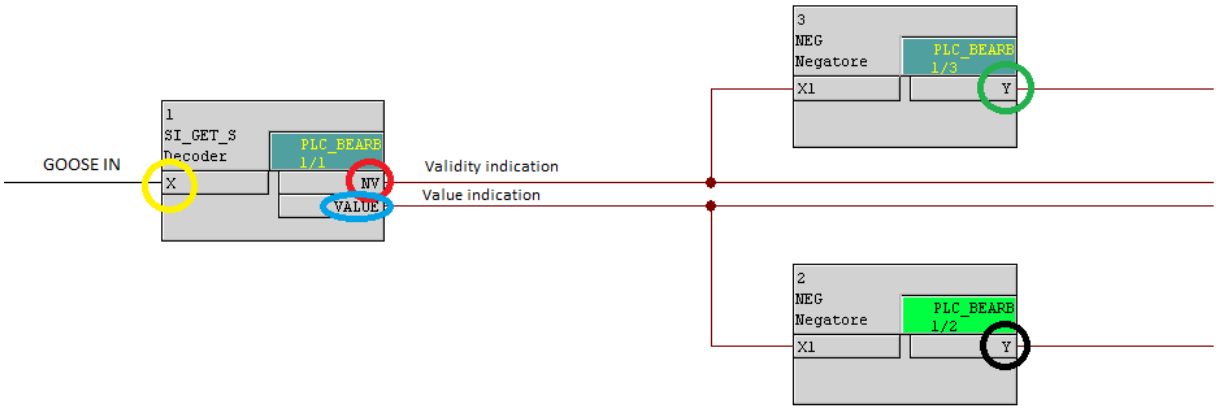


Figure 101 GOOSE message validation logic

If a communication error is revealed, the receiver must switch to a safe state (e.g. acting on its output relays).

To keep track of IED's actions, the receiver is configured to send another GOOSE message each time a communication error is reported, then such a GOOSE message is captured through the PC.

Finally, another GOOSE message is configured to keep track of whether the incoming GOOSE message has been acknowledged or just ignored.

GOOSE communication is handled by IEC 61850 Station and it is necessary to access the system interface to configure the three kind of GOOSE messages explained above (Figure 102).

Summarizing the GOOSE message configuration, three types of messages have been configured.

One that is actually sent by the sender (Type 1) and two that are sent by the receiver: one each time a communication error is revealed (Type 2) and another every time a new GOOSE message is acknowledged (Type 3).

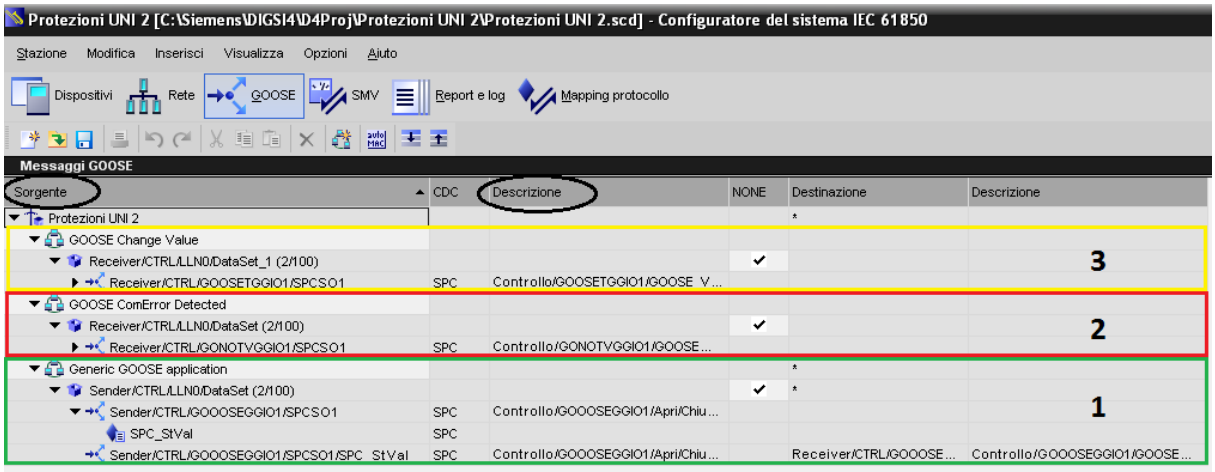


Figure 102 GOOSE configuration

The first kind of messages (green square) represents a generic GOOSE communication from sender to receiver. It refers to the parameter SPCS01 stored in LN GOOOSEGGIO1, the receiver subscribes the exact same GOOSE and it stores the information in the same LN at receiver side.

The second type of messages (red square) is used by receiver to display that a communication error has occurred. In order to do so, another SPCS01, linked to the validity indication and stored in GONOTVGGIO1, is defined.

Lastly, the third class of messages is connected to the actual value of the information carried by each single GOOSE message. Each time this value switches from TRUE to FALSE (and vice versa), a GOOSE message is sent by the receiver. Practically these messages indicate that a new valid GOOSE message has been acknowledged.

For each kind of message, it is possible to define T_{min} as well as T_{max} as the minimum and maximum interval of time between messages, respectively, and the IEEE 802.1Q VLAN Tagging.

A Notebook acts as an Ethernet sniffer which captures every single packet passes through the shared media, and is used to capture the three kind of messages listed above.

3.3.1 Practical configuration

The configuration outlined above, which is considered the basic configuration from which it is possible to extract all the practical configurations, is not adequate to perform the necessary communication tests. All the IEC 61850 IEDs available on the market do not permit a specific manipulation of GOOSE message parameters, such that it is impossible to intentionally create a GOOSE message containing a specific communication error.

To address such an issue, the sender is replaced with another PC equipped with the specific software listed below (Device 1). In this way, the PC is not just able to faithfully replicate the behaviour of a real IEC 61850 IED, but also it allows packet manipulation to create each of the communication error defined in IEC 61784-3.

To do so a specific packet crafting technique is used, where four phases can be highlighted: packet assembly, packet editing, packet play, and packet decoding. Packet assembly consists of the creation of the GOOSE messages to be sent. In packet editing, one or more telegrams are modified in order to create a specific

communication error. Packet play phase is used to send such messages over the Ethernet, while packet decoding determines the targeted network's response to the scenario created by packet play.

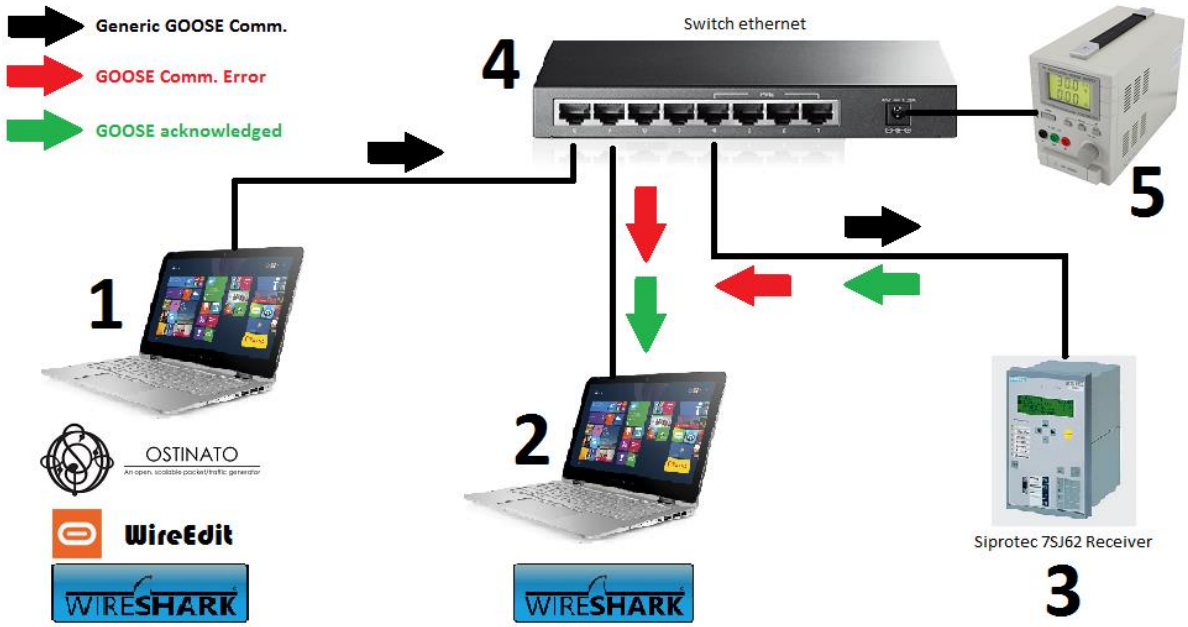


Figure 103 Practical configuration

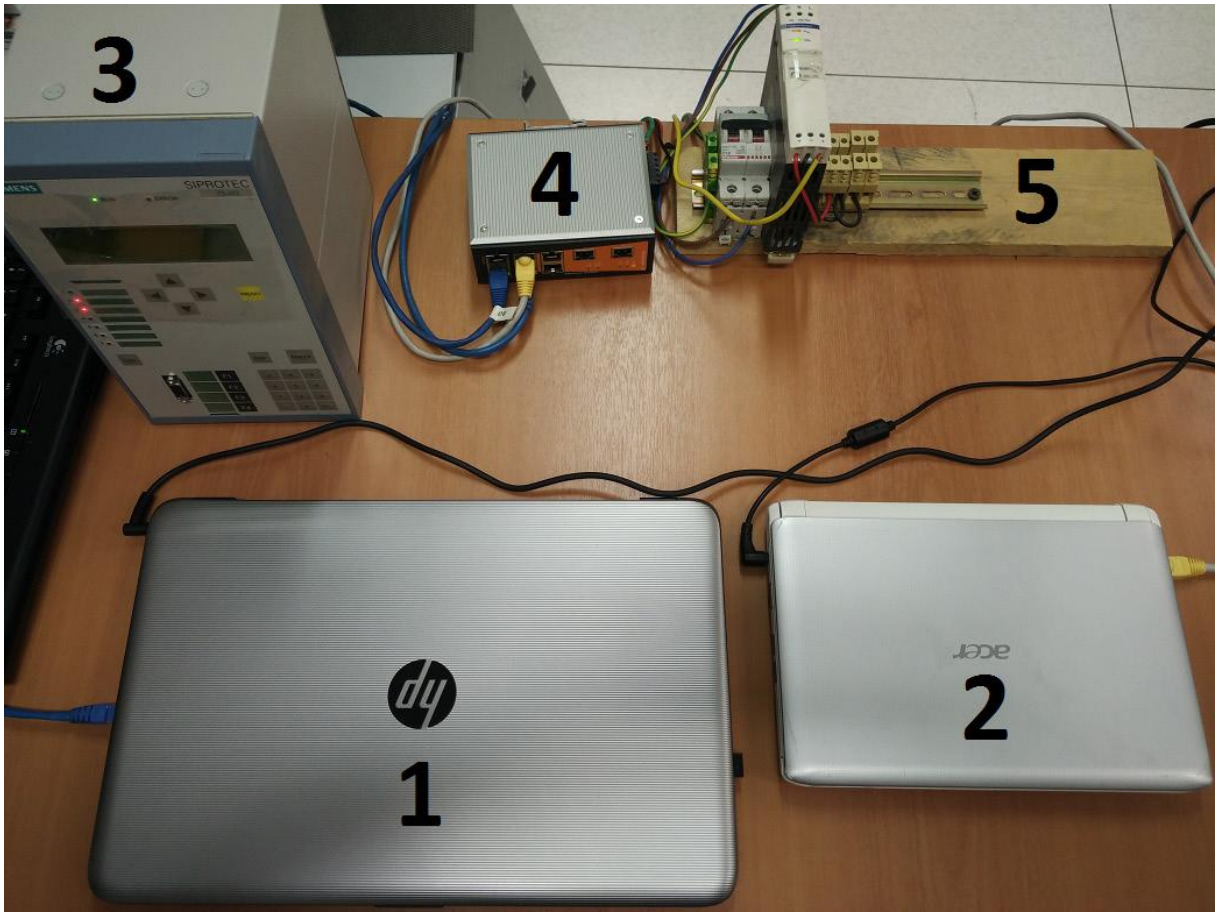


Figure 104 Practical configuration test set

The first device is a Notebook and it is equipped with the three open source software listed below:

- Wireshark: it is the world's foremost and widely-used network protocol analyser, it is used to sniff messages exchanged between devices;
- WireEdit: it is a full stack network packets editor which allows editing packets data at all stack layers (packet editing phase);
- Ostinato: it is a Network traffic generator and analyser, it is used to generate a valid GOOSE traffic and to send manipulated GOOSE messages (packet play phase).

The second device is equipped with Wireshark. This second device is needed because it stays in the same VLAN as Device 3 (SIPROTEC) and its purpose is to listen to what the IED receives and sends.

The third device is the SIPROTEC 7SJ62 previously outlined.

Before moving forward with the actual test, a valid GOOSE message model needs to be provided to the Device 1, so that it can behave as a real IED. In order to do so, for the very first part of this test, the IED was configured as Sender and its packets sniffed through Wireshark (GOOSE model creation will be completely explained later). For the remaining of this practical test, the SIPROTEC device remains configured as Receiver.

The fourth device is a managed switch and a fully compliant IEEE 802.1Q network is created as listed below:

- VLAN ID 1: VLAN only for management purposes;
- VLAN ID 2: VLAN only for safety related devices;
- VLAN ID 4: VLAN for non-safety related devices.

The actual physical ports were configured as follows:

- Port 1: for management purposes, belonging to VLAN 1;
- Port 2 and 3: for safety related devices, belonging to VLAN 2;
- Port 4: Trunk port which accept VLAN Tagged traffic from VLAN 1, VLAN 2 and VLAN 4;
- Port 5: for non-safety related devices, belonging to VLAN 4.

3.3.2 GOOSE message model

As explained above, the sender of GOOSE messages is not a real IED but it is a PC configured to behave in the exact same way as such. Additionally, it provides a way to modify messages in a manner that can replicate communication errors, therefore it is possible to see which the real behaviour of the Receiver is if a communication error has been detected.

The model is created from the captured packets sent by the Sender, then, each single packet is modified through WireEdit to create a sequence of twenty valid GOOSE messages.

The modified fields are shown in Figure 105 and Figure 106.

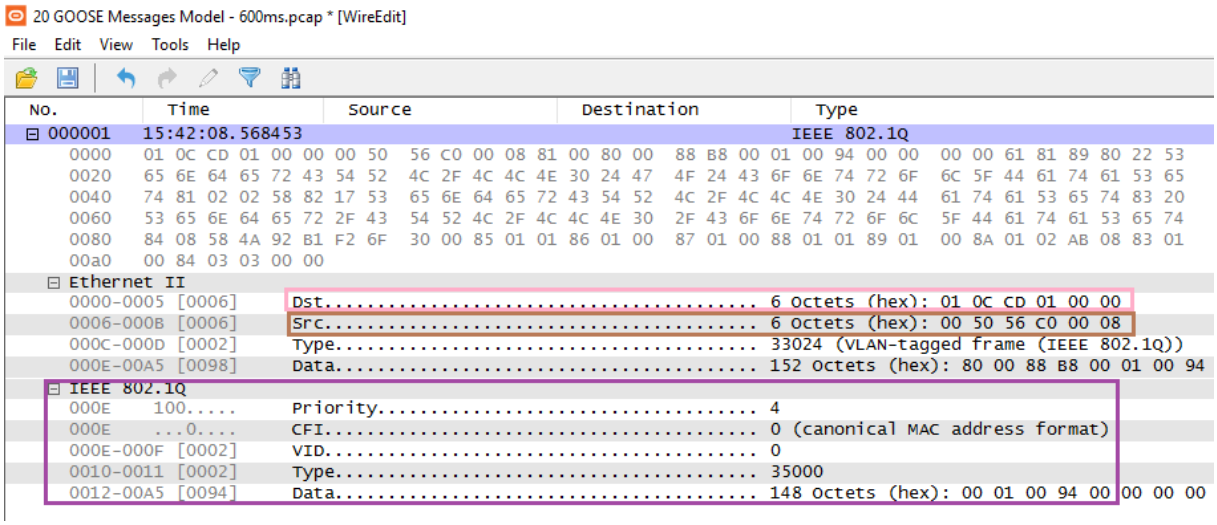


Figure 105 Modified Ethernet parameters

These fields include:

- Source MAC address: its value was put equal to 00:50:56:c0:00:08 and, unless otherwise stated, this MAC identifies a safety-related source address (brown square in Figure 105);
- Destination MAC address: its value was always put equal to 01:0C:CD:01:00:00, it represents the multicast address for IEC 61850-8-1 GOOSE Type 1/1A (pink square in Figure 105);
- VLAN Tag: for this model, VLAN ID=0 and Priority=4 were considered (violet square in Figure 105);
- TimeAllowedtoLive: it includes the TAG, its length and the actual value (yellow square in Figure 106). For this purpose, its value was always considered equal to 600ms for every message;
- TimeStamp: it is composed by 8 bytes and it includes respectively a TAG, its length and the actual value (blue square in Figure 106). For this purpose, unless otherwise stated, the UNIX value equal to Dec 9, 2016 11:17:05 UTC was considered for every message;
- State Number: it includes respectively a TAG, its length and the actual value (red square in Figure 106). This model starts with state number equal to 1 and, unless otherwise stated, it keeps this value constant;

- Sequence Number: it includes respectively a TAG, its length and the actual value (green square in Figure 106). This model starts with sequence number equal to 0 and, unless otherwise stated, Sequence Number always follows an increasing integer sequence (in this case, from 0 to 19);
- Data Value: it is the effective value of the SPS carried by the message, it is equal to 0 unless otherwise stated (black square in Figure 106).

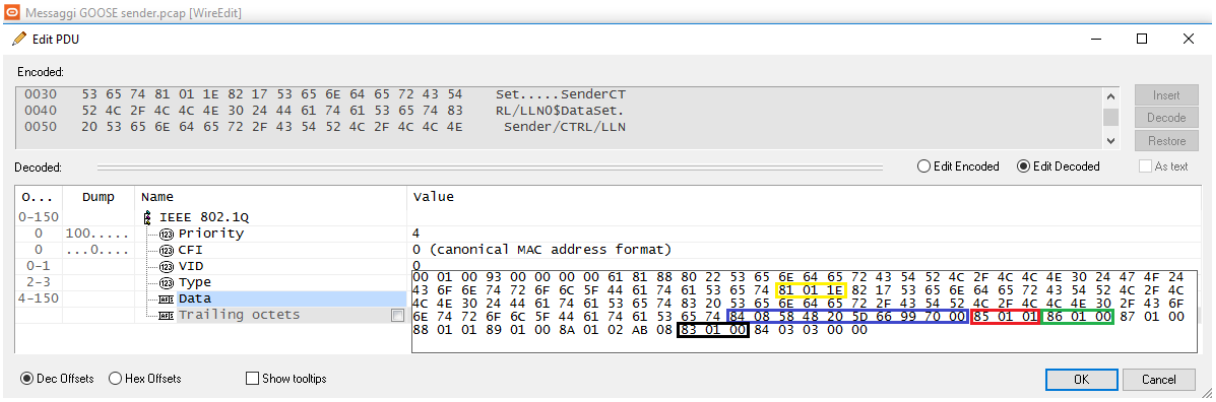


Figure 106 Modified GOOSE parameters

Once each single Ethernet packet has been modified according to the model's needs (packet editing), the .pcap file containing all the modified packets is imported to Ostinato and the sending sequence is properly configured (packet play).

Thus, Ostinato can fully reproduce the exponential sending mechanism of GOOSE messages if correctly configured, for the purpose of this test, a constant sending mechanism equal to 500ms is considered appropriate (TAL constant equal to 600ms). This choice does not involve any changes on the test's results because it is completely up to the sender to decide how quickly and which TAL those messages are sent with. The TAL simply needs to be a value in accordance with the sending speed.

The result is a model composed by twenty valid GOOSE messages, all sent in the correct order and with a fixed speed. All the features explained above are contained in each single GOOSE message; for instance, the Sequence Number assumes a value from 0 to 19 and it displays the sending sequence in which each message belongs.

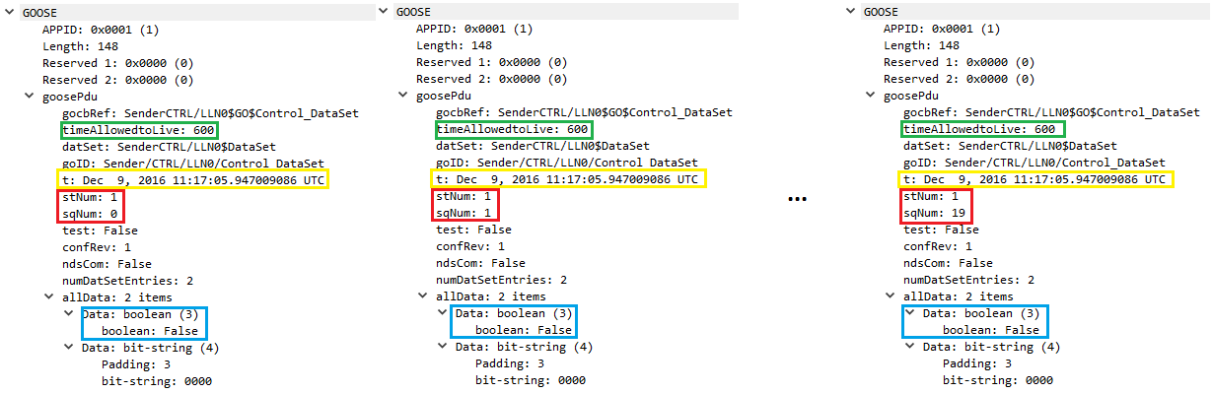


Figure 107 Sequence of 20 GOOSE messages

Such a model is now ready for packet play phase and to be sent through Ostinato. Every IED compliant with IEC 61850 does fully acknowledge each single message, thus considering it as a valid GOOSE message sent by a real IED.

3.3.3 Device's behaviour

Starting from this model, this chapter aims to determine how an IEC 61850 device behaves when a communication error has occurred.

In order to do so, the normal behaviour of each single equipment configured in the test network must be defined.

3.3.3.1 SWITCH ETHERNET (DEVICE 4)

The implemented features are summarized below:

- The Ethernet switch was configured to accept VLAN tagged traffic only at its trunk port (in this case Port 4), all IEEE 802.1Q Ethernet packet incoming from any other port is discarded;
- Ports 2 and 3 connect devices into VLAN 2, so that only safety related devices should access to these ports;
- Port 4 is the trunk port, so that only IEEE 802.1Q tagged traffic is allowed to pass through it; all incoming untagged packets are discarded;
- VLAN 4 is used to connect non-safety-related devices, so that such devices maybe connected to the appropriate port (Port 5 in this case);

- Web interface is accessible only from VLAN 1, so that only port 1 allows access to the switch's configuration.

The actual Ethernet layout is shown in Figure 108.

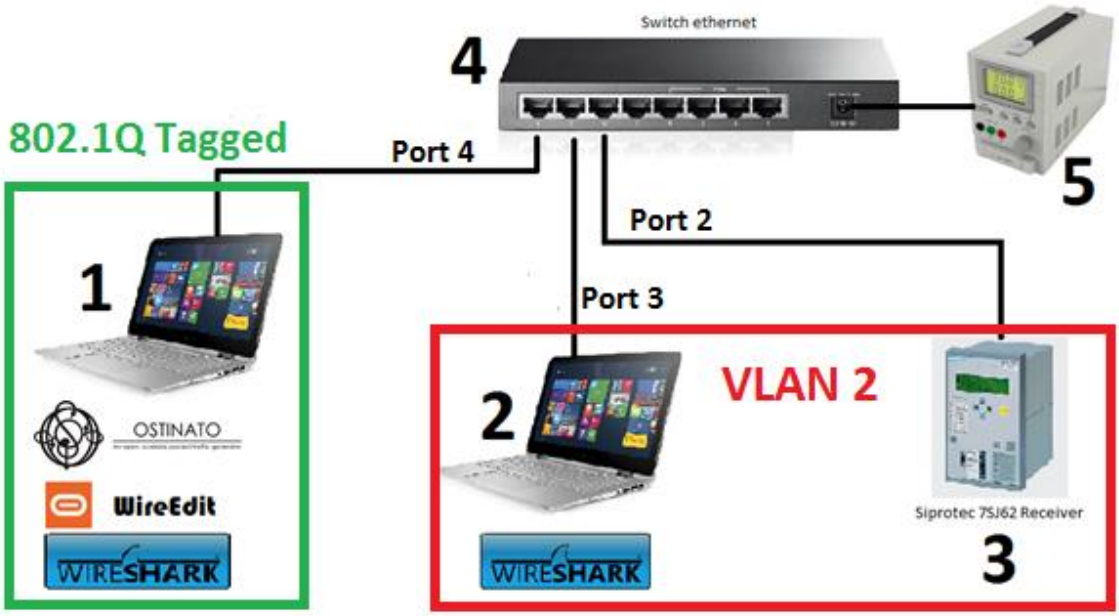


Figure 108 VLAN configuration

3.3.3.2 NOTEBOOK IN VLAN 2 (DEVICE 2)

This laptop has Wireshark running on its Ethernet interface, it listens to exactly the same packets of the IEC 61850 device (Device 3).

It aims to understand which packets are sent and received by Device 3 during packet play phase, in order to fully understand its behaviour in relation to the packets it receives.

3.3.3.3 NOTEBOOK CONNECTED TO TRUNK PORT (DEVICE 1)

This Notebook uses Ostinato to send GOOSE messages over the Ethernet. Each time a communication test is performed, packets that have been modified are identified to create a specific communication error according to IEC 61784-3.

Each time a safety related IED is simulated, Ethernet packets are sent with VLAN ID 2 and, unless otherwise stated, this is the default configuration. To simulate a non-safety related device, VLAN ID 4 is used. In order to simulate traffic from safety and non-safety related devices incoming on trunk port, Ostinato allows to configure a specific VLAN ID for each GOOSE message.

3.3.3.4 IEC 61850 IED (DEVICE 3)

The IED is configured to subscribe GOOSE number one and to publish GOOSE numbers two and three, as defined in 3.3.

Regarding the subscribed GOOSE, it is nothing more than the GOOSE model as defined before, so that, roughly speaking, the SIPROTEC just expects to receive its valid GOOSE message before the TAL expires.

If no further changes are made to the GOOSE model, it contains twenty valid messages ready to be subscribed, each of which is sent every 500ms.

Speaking about the published messages, two types of GOOSEs are defined. The first kind of published messages is sent each time a valid GOOSE message is not received within the correct time window defined with TAL value. Every message contains a TAL field which indicates the allotted time window for the following GOOSE to be considered as valid.

As seen in 3.2.2.1, when a GOOSE contains a parameter that is different from the expectation, the message is ignored with no further indications. If another valid message does not arrive within the same allotted time window, the device assumes that a communication error has occurred and countermeasures must be taken. Additionally, SIPROTEC 4 devices IED tolerates one missing telegram long as the next telegram is received within the time allowed to live time out detection.

Basically, when a generic communication error occurs and a valid message is not received, the error is reported after 2*TAL.

When a communication error is detected, the IED is programmed to publish a GOOSE message, so that its behaviour can be seen through Wireshark. In an actual context, it will reasonably act by means of its output relays but, to be able to report its exact behaviour (in addition to its reaction time), it was programmed to send another GOOSE message which can be captured through an Ethernet sniffer. In this way, a heartbeat GOOSE is replied every 2000ms and, if a communication error is detected, a new GOOSE message is published instantly.

The heartbeat GOOSEs can be distinguished from detection GOOSEs looking at their Data and Sequence Number value. For heartbeat GOOSEs, the carried Boolean value is set to FALSE because no error is detected, otherwise, for detection GOOSEs, it is equal to TRUE.

Figure 109 shows the differences between a heartbeat GOOSE and an error GOOSE, as it can be seen, Data value is changed as well as Time Stamp, State Number and Sequence Number indications.

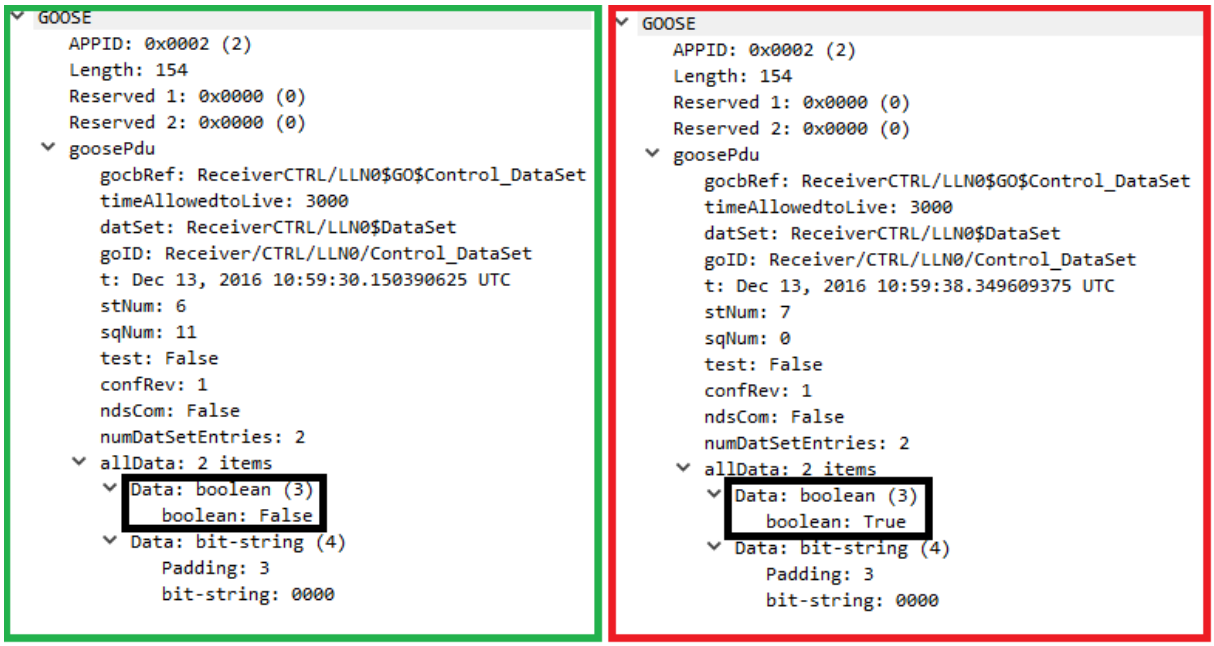


Figure 109 Heartbeat GOOSE vs Error GOOSE

Figure 110 shows a complete GOOSE communication with heartbeat GOOSE messages (green square) repeated exactly every 2000ms and error GOOSE messages (red square) repeated with exponential mechanism.

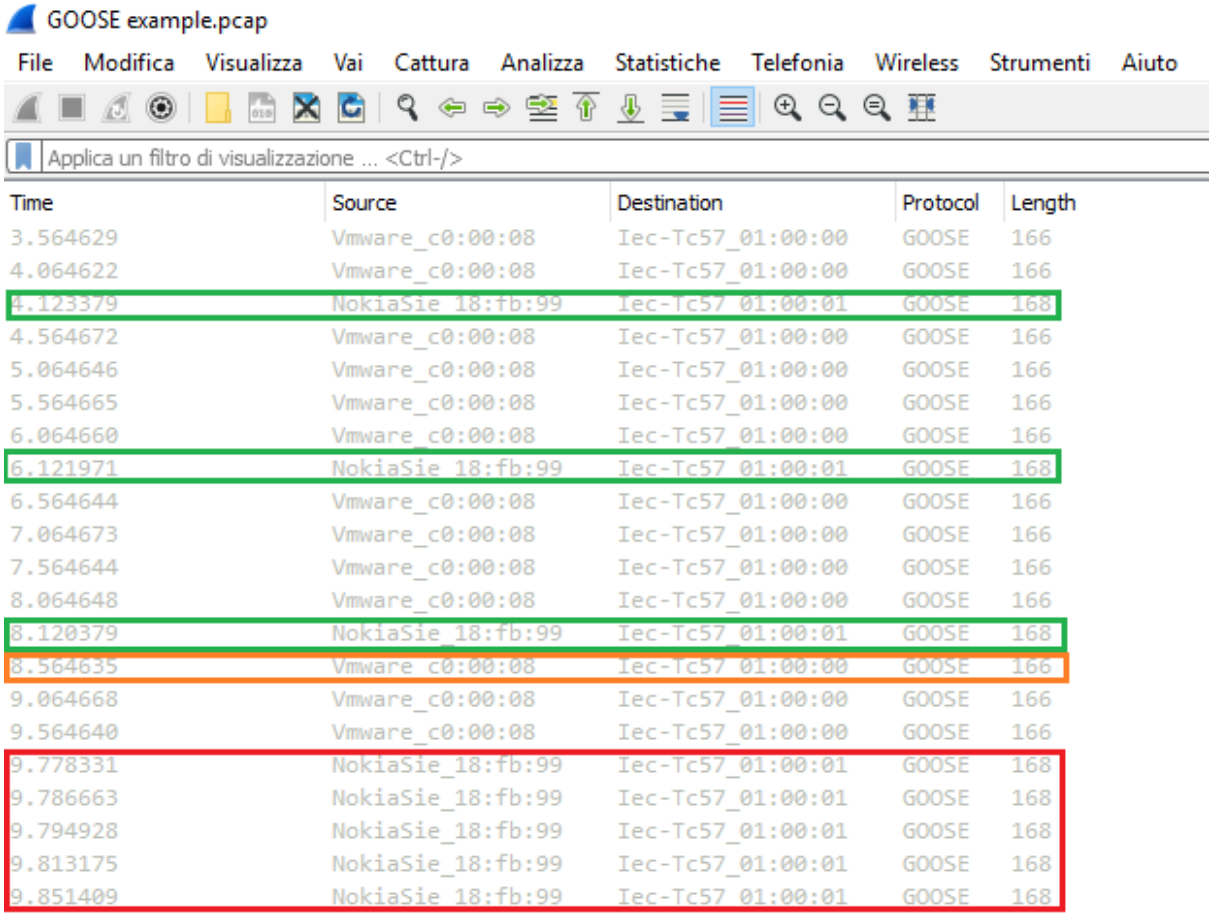


Figure 110 Complete communication with heartbeat GOOSE messages and error GOOSE messages

It is worth to mention that, in this specific example, the last valid packet received is the orange one and the first error GOOSE is sent exactly after 1200ms (2*TAL) from the last valid GOOSE received.

Each time a new GOOSE message is acknowledged, SIPROTEC is configured to respond with another GOOSE message. With this additional GOOSE, in case of ambiguous situations, the messages that have been acknowledged or not can be easily discerned.

These messages have the same mechanism of error GOOSEs as explained above; they are normally sent with a heartbeat mechanism until a new GOOSE message is subscribed. Additionally, they contain a Data value linked to that of the subscribed GOOSEs, so that, if the subscribed GOOSE has a Data value equal to TRUE, the Data field of these messages is TRUE and vice versa.

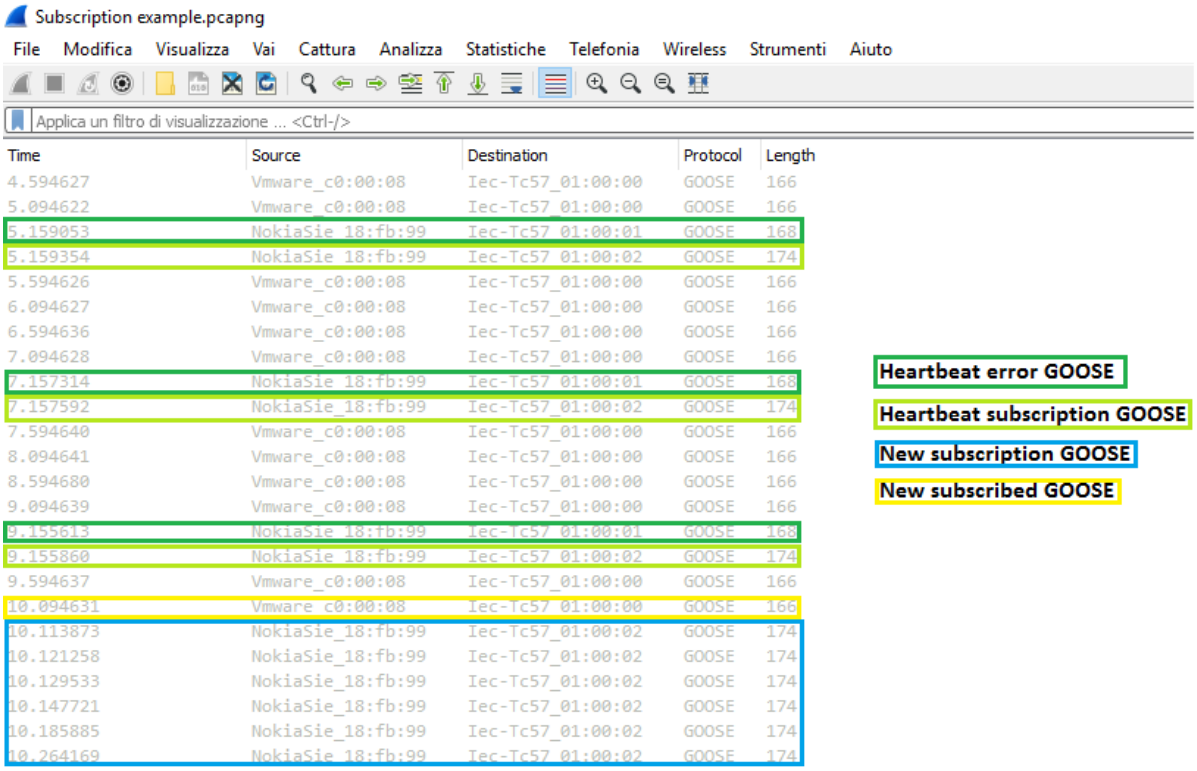


Figure 111 New subscription GOOSE mechanism

Figure 111 shows a GOOSE communication with both types of described GOOSE messages, respectively. Green is for heartbeat error GOOSE messages and light green is for heartbeat subscription GOOSE messages. Each time a new GOOSE message is subscribed (yellow square), a new GOOSE message exponential sending mechanism starts (blue square).

A detailed view of those messages is provided by Figure 112.

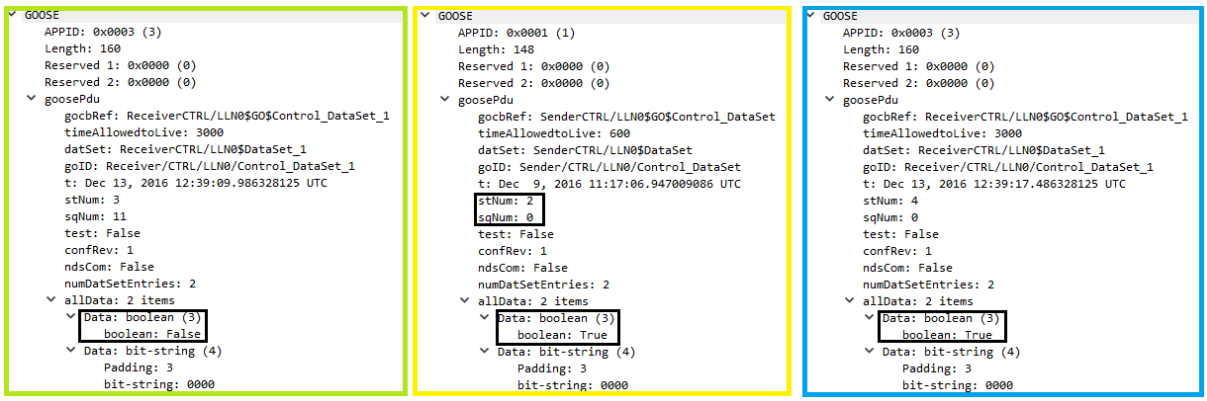


Figure 112 Heartbeat subscription GOOSE, new subscribed GOOSE message, new subscription GOOSE message

An actual example of a complete GOOSE communication is provided by Figure 113, it contains all the three types of configured GOOSE messages as well as heartbeat and exponential retransmission mechanisms.

It is worth mentioning that the time elapsed between the sending of a new valid GOOSE message (yellow square) and the reception of the response from the IED (blue square) is about 20ms for these devices.

Usually, as all GOOSE messages are sent through multicast mechanism, a generic IED may receive a GOOSE message which that specific IED is not supposed to subscribe, for example a message designated to another IED which belongs to the same VLAN network. In this case, as the IED is not programmed to subscribe that specific message, the incoming GOOSE is received but it is not subscribed, i.e. it is simply ignored.

To see whether the incoming message is potentially available for subscription or not, among other parameters, the GoID, gocbRef and dataSet fields needs to be compliant with those the IED is allowed to subscribe.

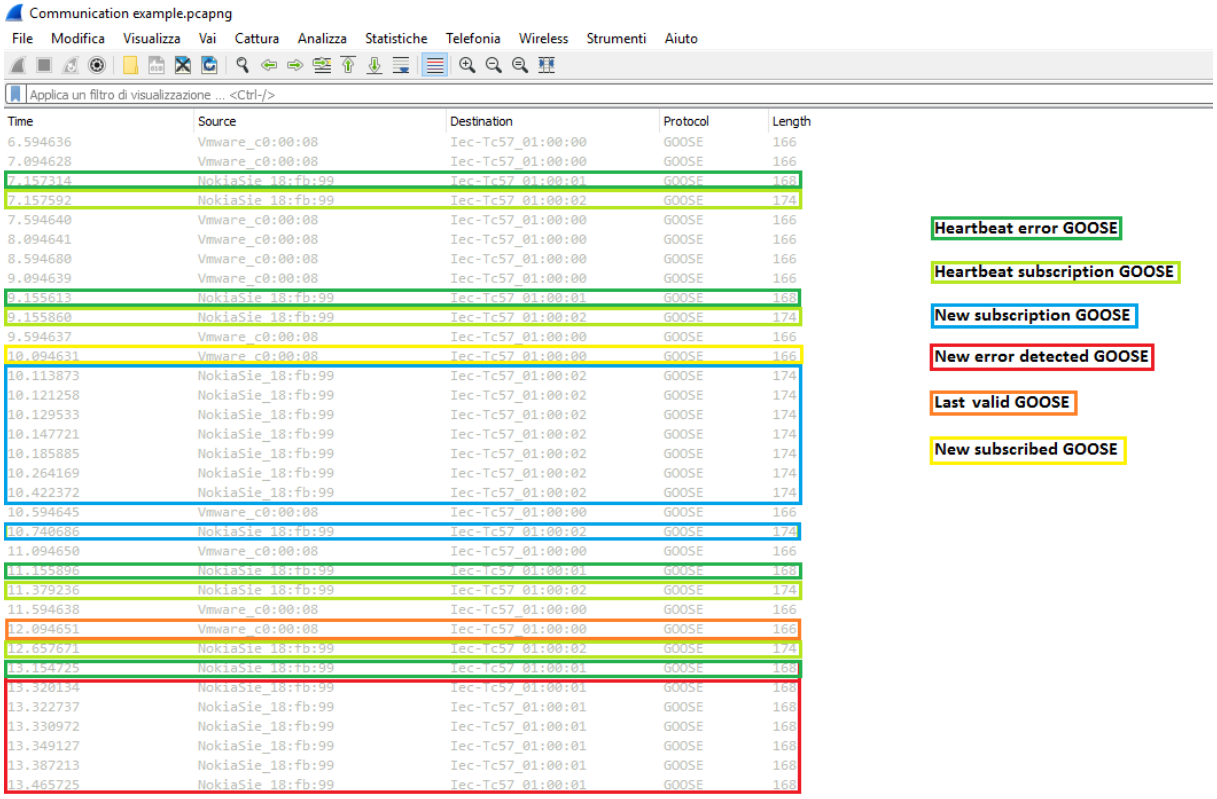


Figure 113 Example of a complete GOOSE communication

3.3.4 Laboratory test

Starting from the GOOSE model explained in 3.3.2, each communication error, as defined in IEC62784-3, is widely discussed and simulated.

For each of the possible communication errors, hardware and software specification is fully provided as well as the way that specific error was implemented inside the model.

Unless otherwise stated, the configuration matches exactly with the one described in 3.3.1 and the default VLAN configuration used is shown in Figure 114.

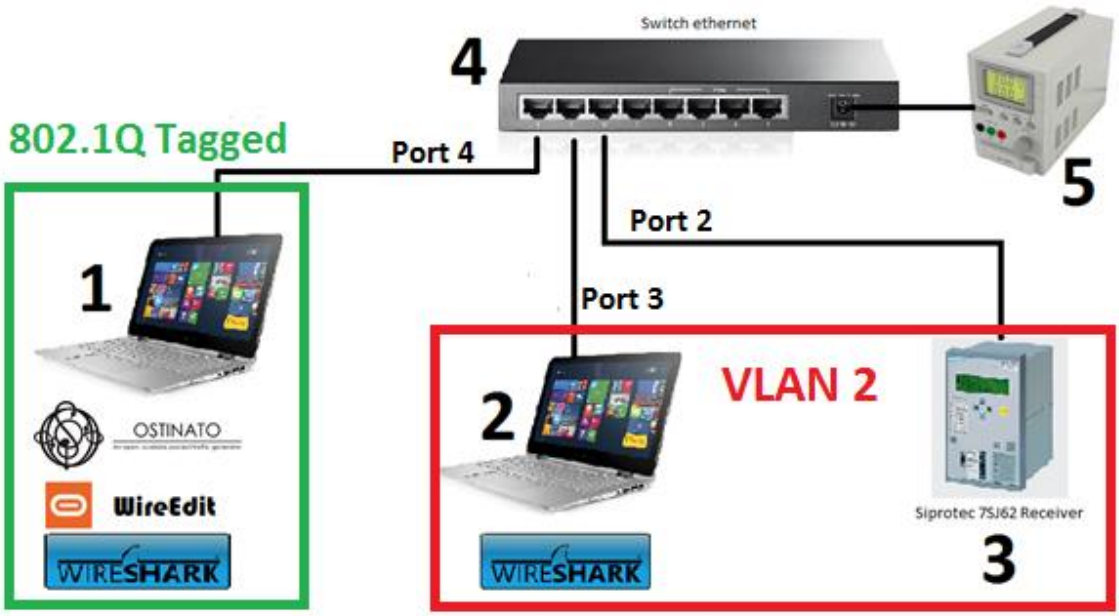


Figure 114 VLAN configuration

For practical reasons, during all the following analysis, any heartbeat and repeated GOOSEs will be ignored, such that the shown messages are the only relevant messages for the purpose of this thesis.

Figure 115 shows the differences between one capture file with heartbeat and repeated messages and the same capture file where those messages are neglected.

In this light view, it is also possible to see the twenty messages the model is composed of; additionally, the first two heartbeat GOOSEs are kept in order to maintain the same time reference.

Unless otherwise specified, all Wireshark capture are made by Device 2 and Ethernet frames sent by Device 1 are always tagged with VLAN ID 2.

To facilitate the reading of the results, these colours will be always used:

- Green: a communication error heartbeat GOOSE
- Light blue: a response GOOSE indicating a new acknowledged GOOSE message
- Light green: a new acknowledged heartbeat GOOSE

- Orange: the last valid GOOSE message that has been acknowledged before a communication error occurs
- Pink: the first GOOSE message, classified as valid, following an error
- Red: a GOOSE indicating that a communication error has occurred
- Yellow: a GOOSE message carrying a new data value

To indicate generic GOOSE messages, colours violet, brown, grey, turquoise and black are used.

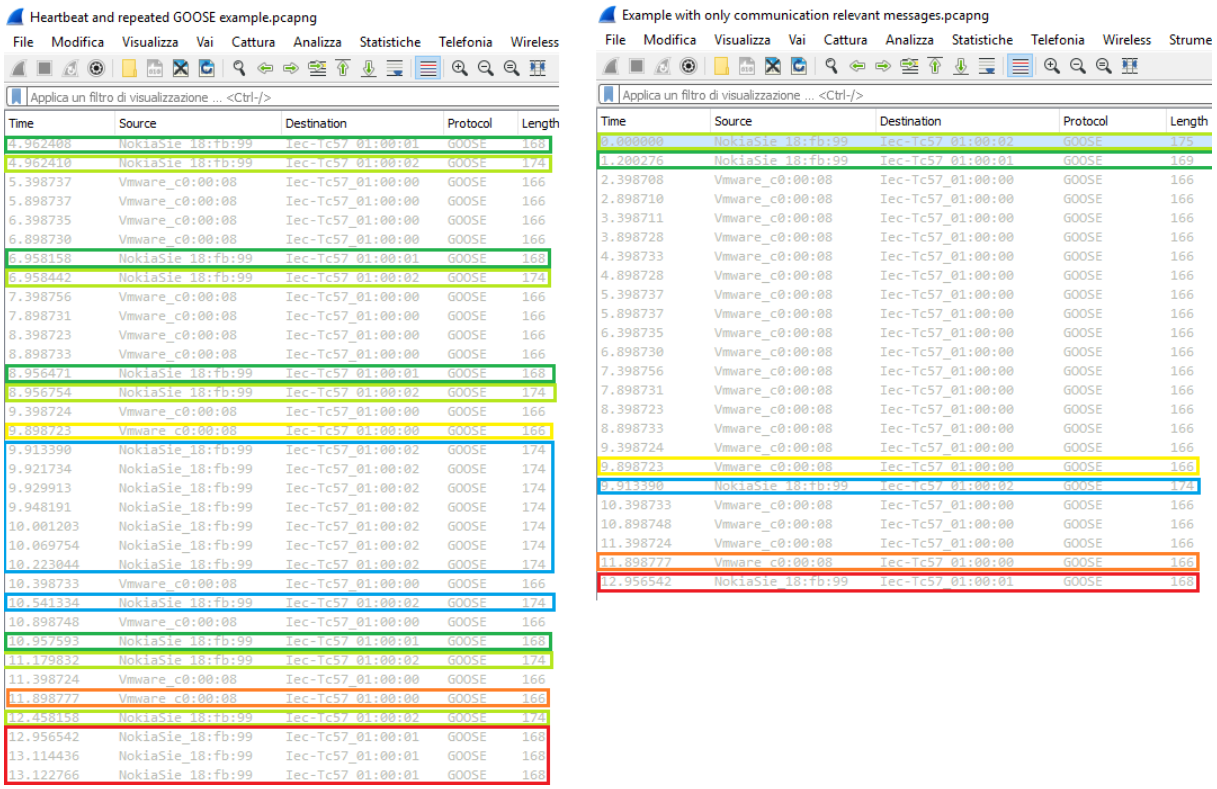


Figure 115 Example of a complete communication and the same one with no repetitions

3.3.4.1 Loss Test

For this specific communication error two tests were done, each of which missed respectively one and two GOOSE messages.

3.3.4.1.1 One packet loss

For the first test, the GOOSE sequence model was deprived of its packet number fifteen.

In order to simulate a real loss, packet sixteen was delayed by 500ms, therefore GOOSE number fourteen and sixteen are separated by 1000ms. In this way, the resulting average sending rate is lower than 2pps.

As SIPROTEC 4 devices tolerate one missing telegram as long as the next telegram is received within 2*TAL, therefore this IED did not report any communication error even though GOOSE message number fifteen was missing (Figure 116).

Specifically, the allowed time window for GOOSEs fifteen and sixteen was 1200ms, the latter arrived 1000ms after GOOSE fourteen, so that it was considered valid and a communication error was not reported.

This result was expected because, as stated in SIPROTEC's features, they tolerate up to one missing packet, if the following arrives within its allotted time window.

A detailed view of the two GOOSE messages is given in Figure 117.

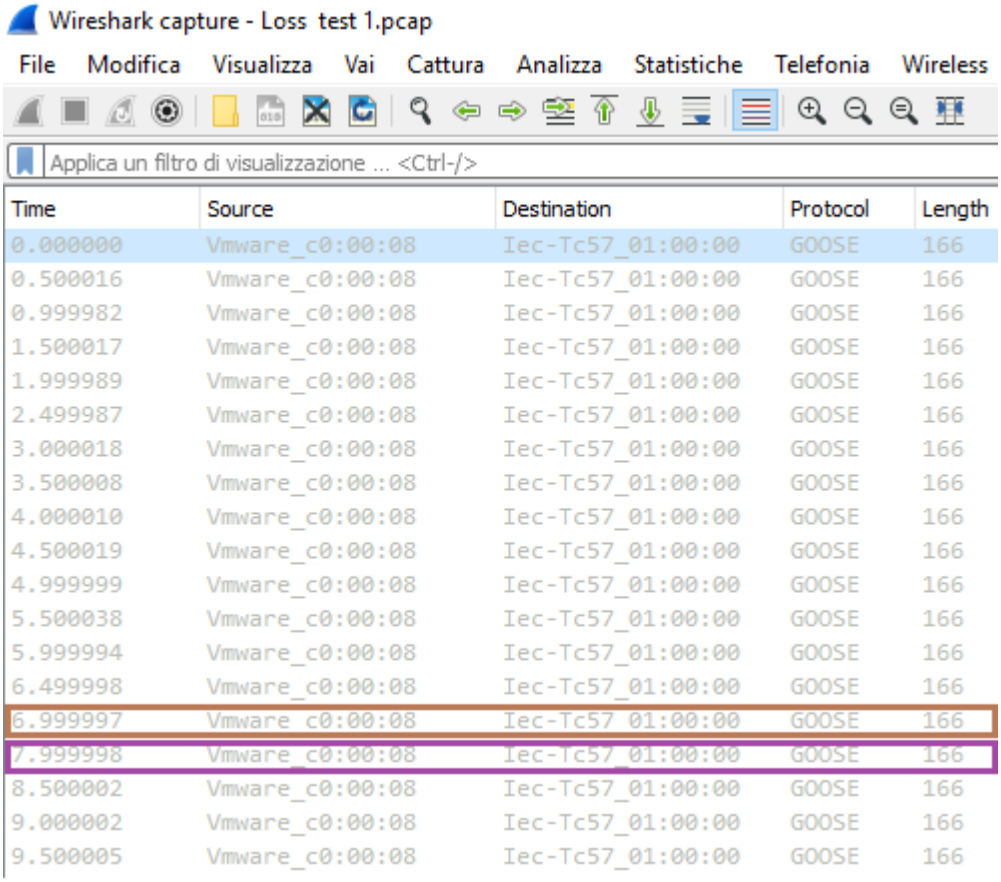


Figure 116 Loss – Test one – Wireshark capture

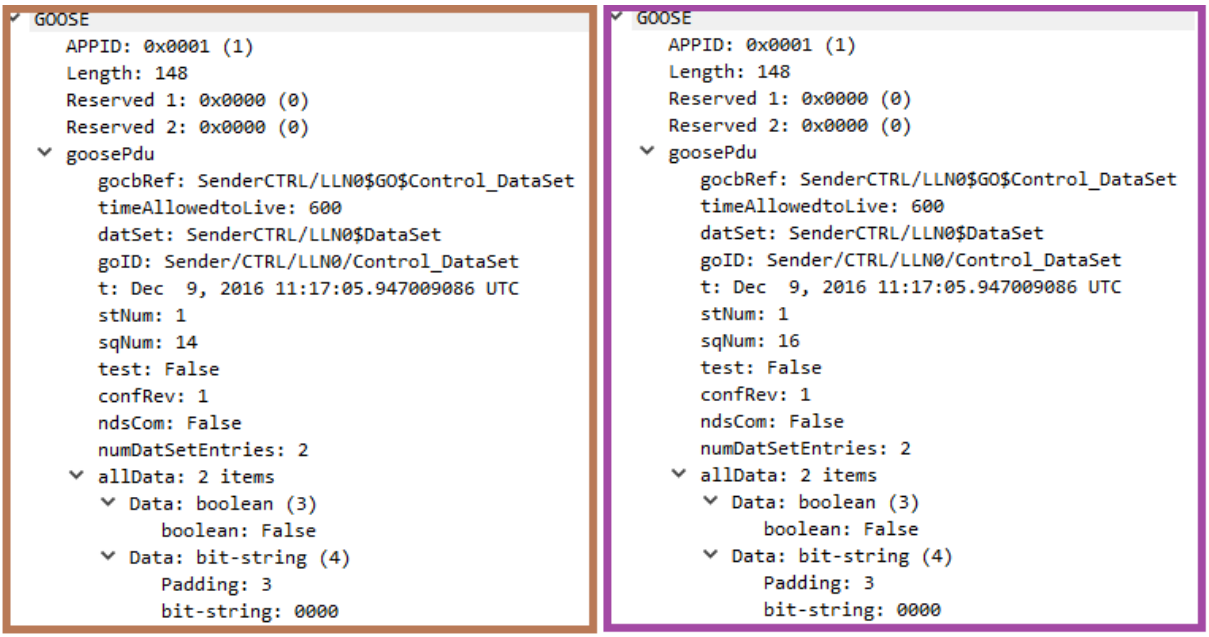


Figure 117 Loss - Test one - GOOSEs detailed view

3.3.4.1.2 TWO PACKETS LOSS

For the second test, the GOOSE model was deprived of its packets number fifteen and sixteen.

In order to simulate a real loss, telegram seventeen was delayed by 500ms, therefore GOOSE number fourteen and seventeen are separated by 1500ms. In this way, the resulting average sending rate is lower than 2pps.

As can be seen in Figure 118, the allotted time window for GOOSEs fifteen and sixteen was 1200ms, but both were removed and message seventeen arrived after 1500ms.

Specifically, the IED expected messages fifteen or sixteen within the TAL that was indicated in GOOSE fourteen. Given that neither the fifteenth nor the sixteenth telegram arrived after 1200ms from the fourteenth message, a communication error was reported.

A detailed view of the three GOOSE messages is provided by Figure 119.

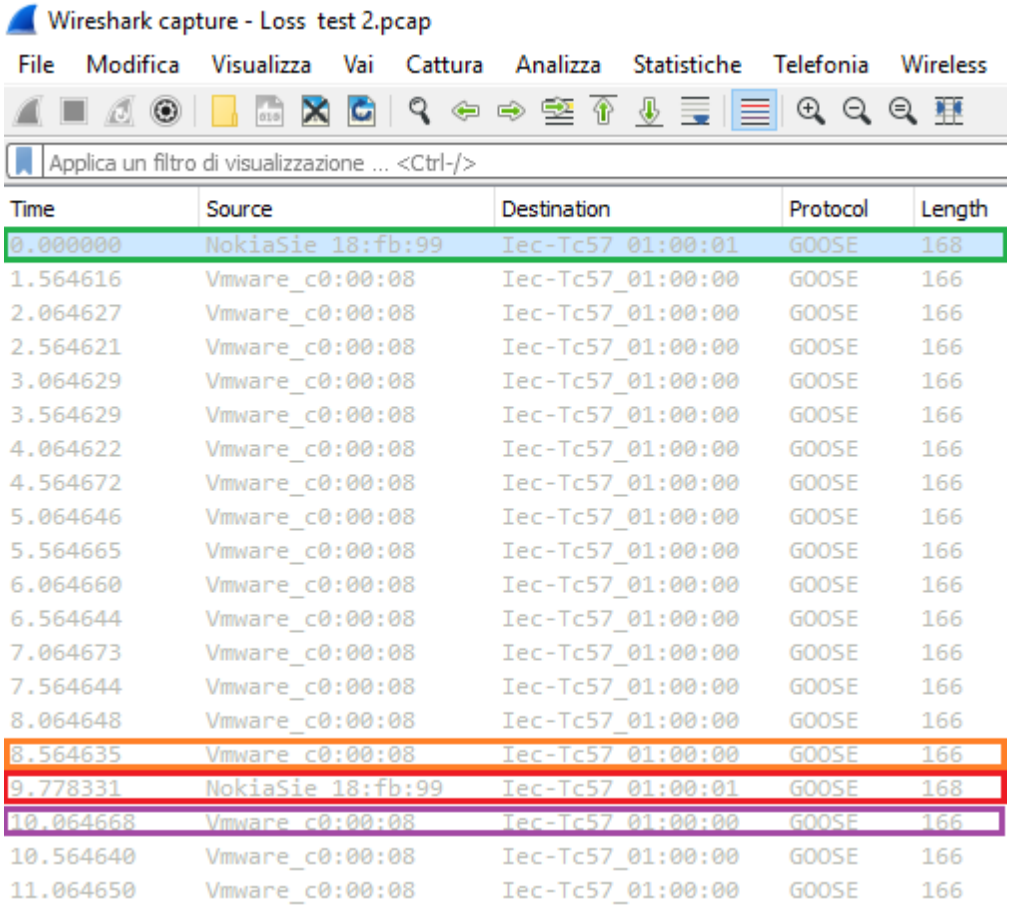


Figure 118 Loss – Test two – Wireshark capture

Figure 119 Goose message detail - Loss test 2

3.3.4.1.3 ADDITIONAL TEST AND CONCLUSIONS

In addition to these two tests above, with the same philosophy as test number two, a supplementary third test was done.

This test aims to see which would have been the SIPROTEC's behaviour if message seventeen would have arrived within the TAL stated in the fourteenth message.

Results in Figure 120 show that if two or more GOOSE messages are lost during a communication, SIPROTEC 4 devices detect and report a communication error. In this case the seventeenth GOOSE (violet square) arrived within the time window allotted for the fifteenth/sixteenth GOOSE and, as the seventeenth telegram was not part of the expected ones, it was ignored. Failing to receive a valid message within the TAL stated in message fourteen, regardless the reception of the seventeenth GOOSE, a communication error was reported when the allotted time window elapsed.

It is completely up to the manufacturer to decide how many losses can be tolerated by their devices. The point is that, depending on the way each device was programmed by the manufacturer, IEC 61850 compliant devices can detect packet losses during a GOOSE communication, for instance a safety-related communication, as defined in IEC 61784-3.

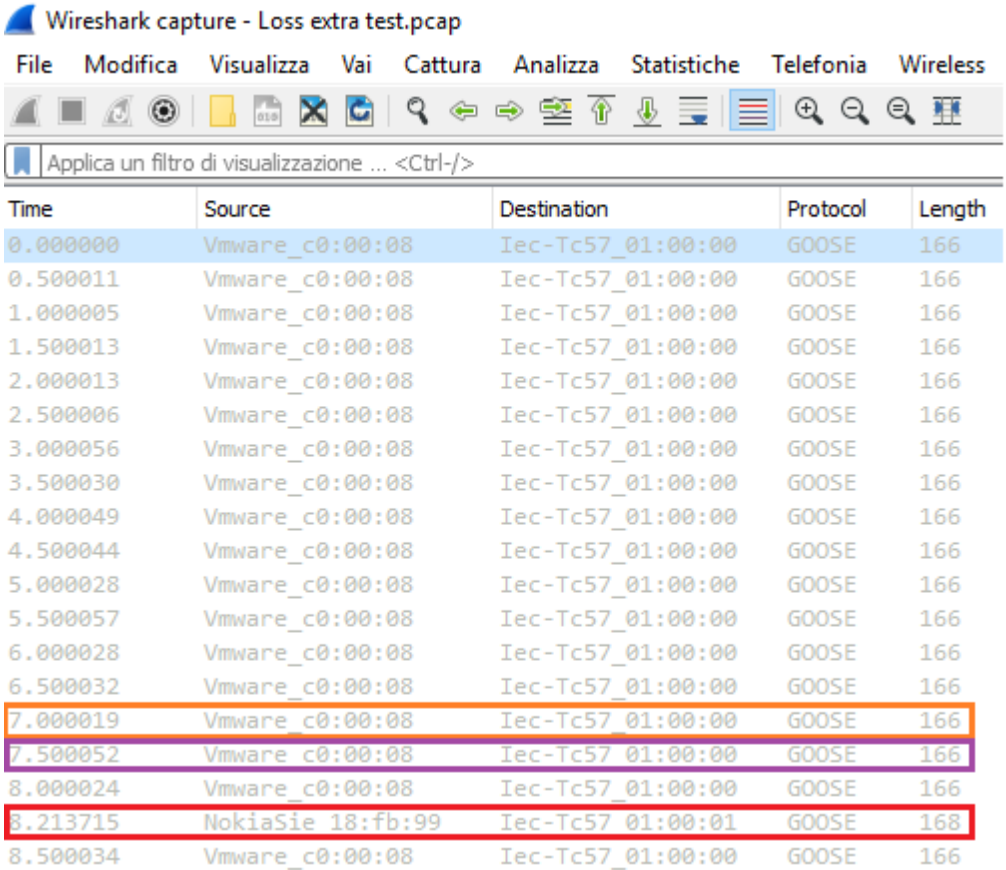


Figure 120 Loss – Additional test – Wireshark capture

3.3.4.2 INCORRECT SEQUENCE TEST

Incorrect sequence communication error occurs each time, due to an error, fault or interference, the predefined sequence (for example natural numbers, time references) associated with messages from a particular source is incorrect.

For this communication error three types of tests were performed, each contributing to the determination of the exact behaviour of an IEC 61850 device in the presence of an incorrect sequence of GOOSE messages.

3.3.4.2.1 INCORRECT SEQUENCE NUMBER

This kind of errors consider only communication errors which involve the reception of GOOSE messages with a different order from the sending one. In order to do so, three

tests with packet order inversion were done, which contain an increasing number of inversions.

The first test involves an incorrect sequence of GOOSEs sixteen and seventeen, particularly their order was exchanged.

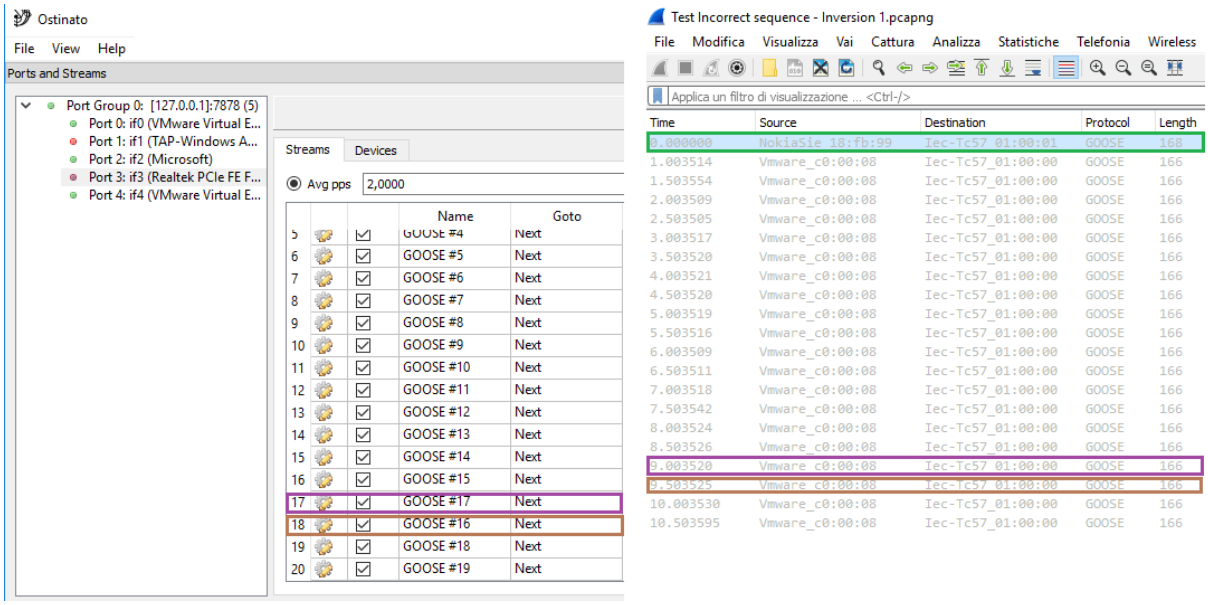


Figure 121 Incorrect sequence - Inversion Test one – Ostinato's GUI on the left and Wireshark capture on the right

As can be seen, despite the incorrect sequence, the SIPROTEC did not report any communication error.

This behaviour was expected because, when packet seventeen was received, it was acknowledged as a valid GOOSE and, at the same time, the sixteenth was marked as lost. Furthermore, GOOSE number eighteen arrived after 1000ms from the seventeenth, therefore the SIPROTEC assumed that a communication error had not occurred and the error was not reported.

The second test involves an incorrect sequence of GOOSEs sixteen, seventeen and eighteen; particularly their order was exchanged.

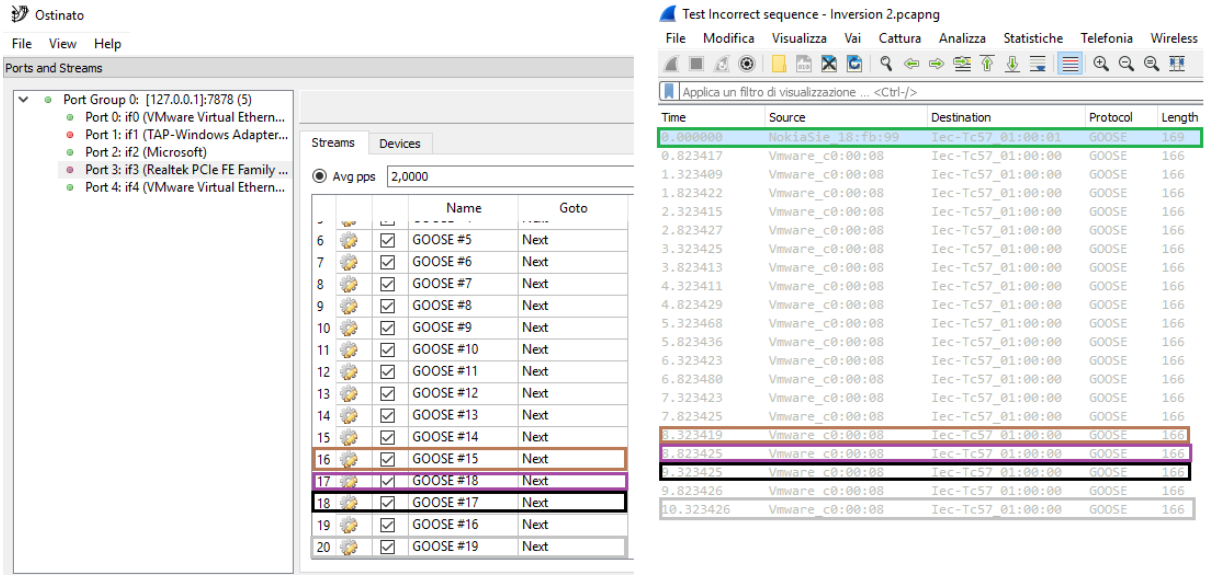


Figure 122 Incorrect sequence - Inversion Test two – Ostinato's GUI on the left and Wireshark capture on the right

Despite the incorrect sequence, the SIPROTEC did not report any communication error.

Below is the explanation of the exact behaviour for each single received GOOSE message:

- GOOSE number fifteen is acknowledged and TAL set to 600ms;
- GOOSE number eighteen is not part of the expected ones, so that it cannot be considered as valid and, therefore, it is ignored;
- GOOSE number seventeen arrives exactly 1000ms after the fifteenth, therefore it is classified as valid and TAL is set to 600ms again;
- GOOSE number sixteen is not part of the expected ones because it refers to an already acknowledged state, therefore it is ignored;
- GOOSE number nineteen arrives after another 1000ms from the last valid GOOSE, it is part of the expected one and, therefore, it is acknowledged.

At this point, if no further error is experienced, the communication can continue without any alarm provided by the IED.

The third test involves an incorrect sequence of GOOSEs fifteen, sixteen, seventeen and eighteen, particularly their order was exchanged (Figure 123).

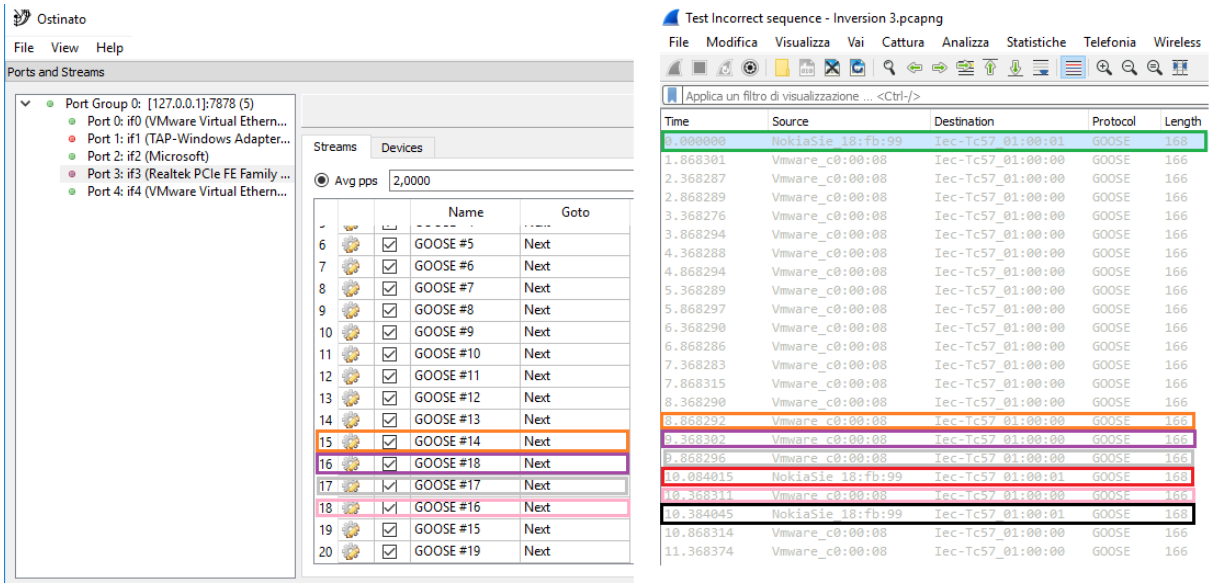


Figure 123 Incorrect Sequence - Inversion Test three – Ostinato's GUI on the left and Wireshark capture on the right

Figure 124 Incorrect Sequence – Inversion Test three - GOOSEs detailed view

In this third test, the communication error was detected and reported by the IED through a GOOSE message at time “10.084015” (red square). The latter contains a data value equal to TRUE indicating that the communication must be considered no more valid.

Below is the explanation of the exact behaviour for each single received GOOSE:

- GOOSE number fourteen is acknowledged and TAL set to 600ms (orange square);
- GOOSEs numbers eighteen and seventeenth are not part of the expected ones, so that they cannot be considered as valid and, therefore, they are ignored (violet and grey square);
- At this point, exactly after 1200ms from telegram fourteen, the time window expires without receiving a valid GOOSE message and an alarm GOOSE, with data value equal to TRUE, is sent (red square);
- GOOSE number sixteen arrives 1500ms after the last valid GOOSE and, as it is part of the expected ones, it is acknowledged;
- Given that the communication must be considered valid again due to the reception of a valid GOOSE, a new alarm GOOSE carrying data value equal to FALSE is sent (turquoise square).

This last test revealed that incorrect sequences in GOOSE messages can be revealed by IEC 61850 compliant devices. In this particular case, the IED detected that no valid packages arrived within the allowed time window, therefore a communication error was assumed and an alarm GOOSE was sent.

3.3.4.2.2 INCORRECT SEQUENCE NUMBER WITH CHANGE OF DATA VALUE

These tests are to be considered as complement of those showed in the previous section.

This section aims to understand whether, the GOOSE messages supposed ignored during the previous section, are really ignored by the SIPROTEC or not.

In order to do so, starting from GOOSE number fifteen, the model was changed as listed below:

- State Number was set equal to 2
- Sequence number started again from 0
- Time Stamp was put one second ahead
- Data value was set equal to TRUE

As the IED was also programmed to send a GOOSE each time a new GOOSE message is acknowledged, this new model allowed to understand exactly which was the first out of ordered GOOSE message considered valid.

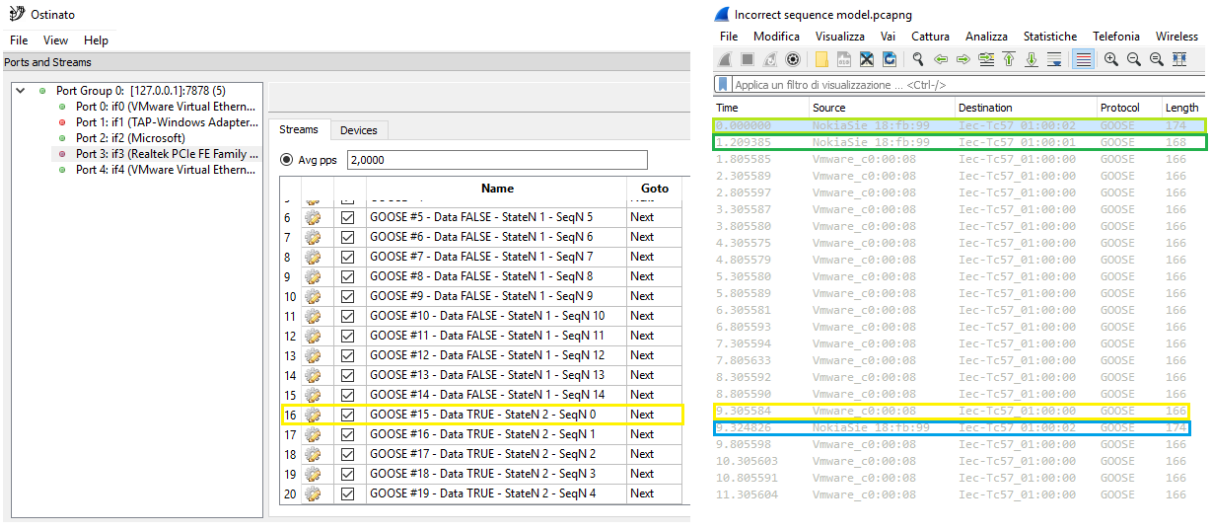


Figure 125 Incorrect Sequence - Data change – GOOSE model on the left and Wireshark capture on the right

The first test was repeated in the exact same way of the previous section; Ostinato's GUI and Wireshark capture are shown in Figure 126.

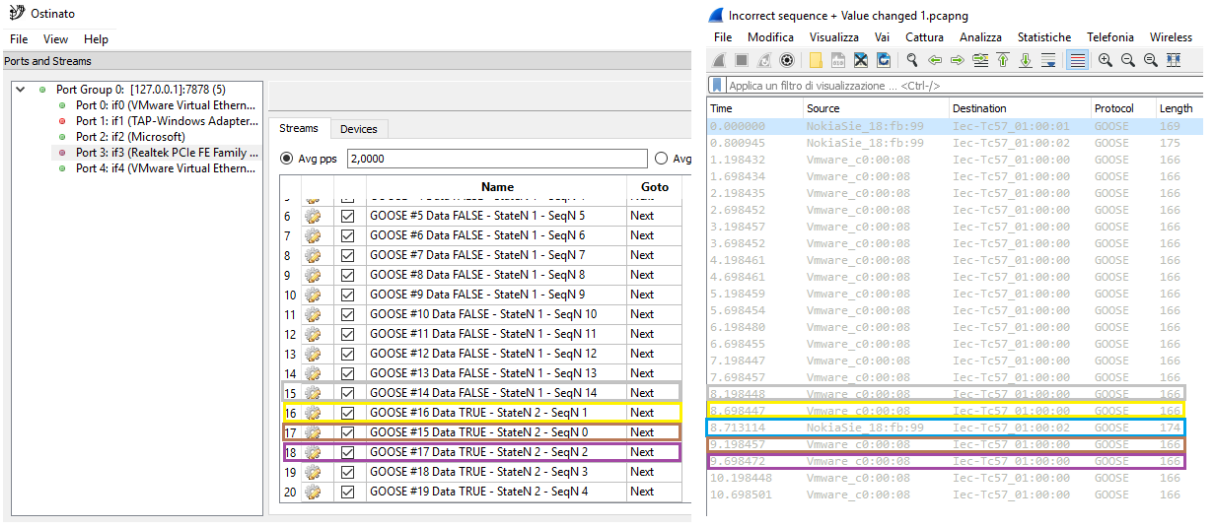


Figure 126 Incorrect Sequence - Data change – Test one - Ostinato GUI on the left and Wireshark capture on the right

This test showed that, despite GOOSE number sixteen arrived out of order, it was acknowledged as valid by the SIPROTEC with no error reported.

Furthermore, telegram with State Number 2 and Sequence Number 0 was ignored by the IED because it referred to a previous Sequence Number.

A detailed view of the affected GOOSE messages is provided by Figure 127.

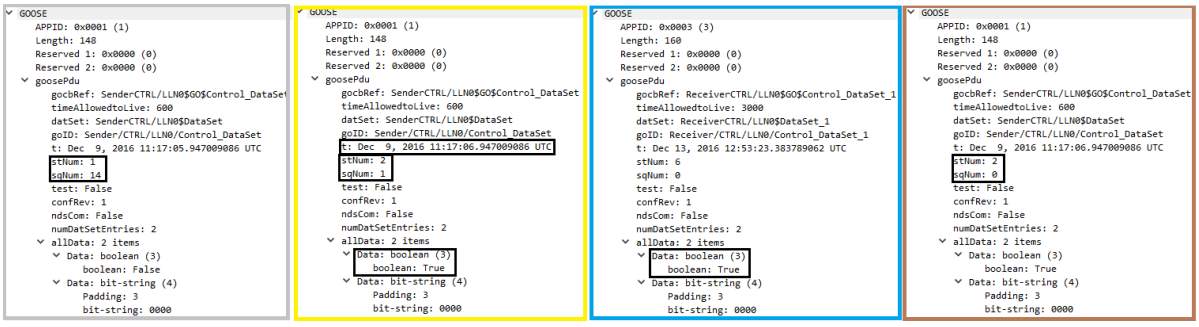


Figure 127 Incorrect Sequence – Data change – Test one - GOOSEs detailed view

Test number two was conducted with the same methodologies of the previous one and no further result was obtained, so it is omitted.

The third and last test follows the same line of that in the previous section, where GOOSEs number fifteen, sixteen, seventeen and eighteen were sent out of order.

Figure 128 shows the exact order in which the messages were sent to the IED and its response.

Below is the explanation of the exact behaviour for each single received GOOSE message:

- GOOSE number fourteen is acknowledged and TAL set to 600ms (orange square);
- As no response from the SIPROTEC is received, GOOSE seventeen and eighteen are ignored by the IED because they are not part of the expected messages (respectively violet and grey square);
- After 1200ms from the last valid GOOSE, TAL is elapsed, a communication error is assumed by the IED and an error GOOSE is sent (red square);
- GOOSE number sixteen arrives 1500ms after the last valid GOOSE and, as it is part of the expected ones, it is acknowledged (yellow square);
- A response message is sent by the SIPROTEC, which demonstrate that all the previews GOOSEs has been ignored.

A detailed view of the affected GOOSE messages is provided by Figure 129.

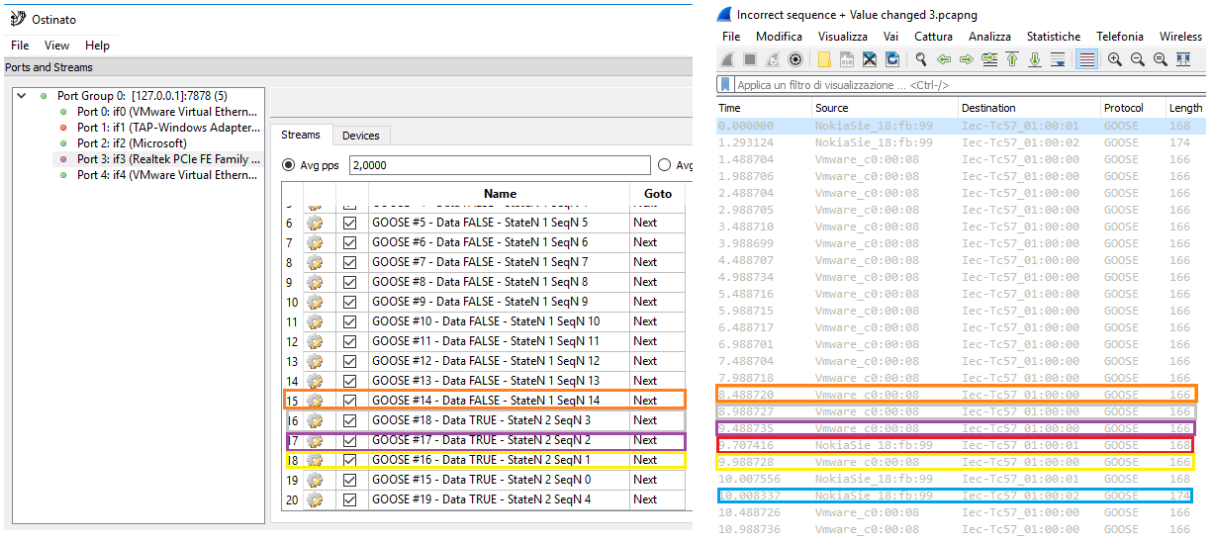


Figure 128 Incorrect Sequence – Data change - Test two – Ostinato's GUI on the left and Wireshark capture on the right

Figure 129 Incorrect Sequence – Data change - Test two - GOOSEs detailed view

3.3.4.3 INCORRECT TIME SEQUENCE

As described before, Siprotec 4 device doesn't process Time Stamp in order to define whether a GOOSE is valid or not.

On this purpose, these tests aim to determinate which is the behaviour of SIPROTEC devices if GOOSEs with incorrect time sequence are received. In order to do so, two tests were done.

During the first test, starting from the fifteenth GOOSE, Time Stamp indication was put 1 second behind while the remaining parameters were left untouched (Figure 130).

As expected, the IED acknowledged the GOOSEs with Time Stamp parameters modified and treated them as valid.

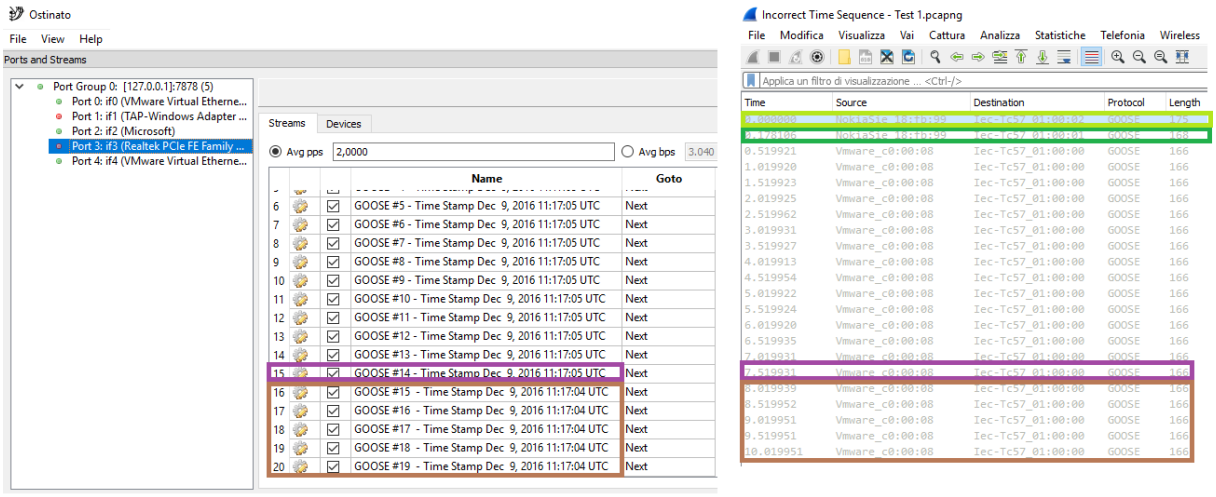


Figure 130 Incorrect Sequence – Time Sequence – Test one - Ostinato's GUI on the left side and Wireshark capture on the right side

The second test is to be considered as complement of previous one.

For this test, in addition to the GOOSEs defined as model, one GOOSE with the following parameters is inserted (yellow square in Figure 131):

- Time Stamp value equal to Dec 5, 1999 16:28:16 UTC
- State Number equal to 2
- Sequence Number equal to 0
- Data value set to TRUE

When the additional GOOSE was received, the IED did not check the Time Stamp indication, so that that message was classified as valid and a confirmation GOOSE was sent (blue square in Figure 132).

In addition to such wrong acknowledgement, the SIPROTEC did not recognize the subsequent messages as valid, leading to a communication error (red square in Figure 131), even though their Time Stamp was ahead to that carried by the additional GOOSE.

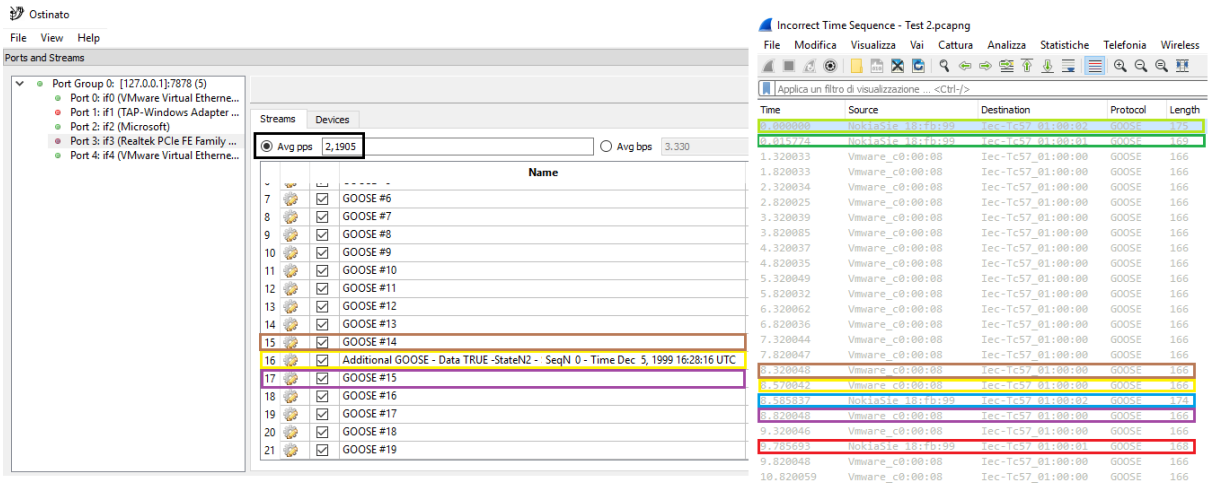


Figure 131 Incorrect Sequence – Time Sequence - Test two – Ostinato's GUI on the left and Wireshark capture on the right

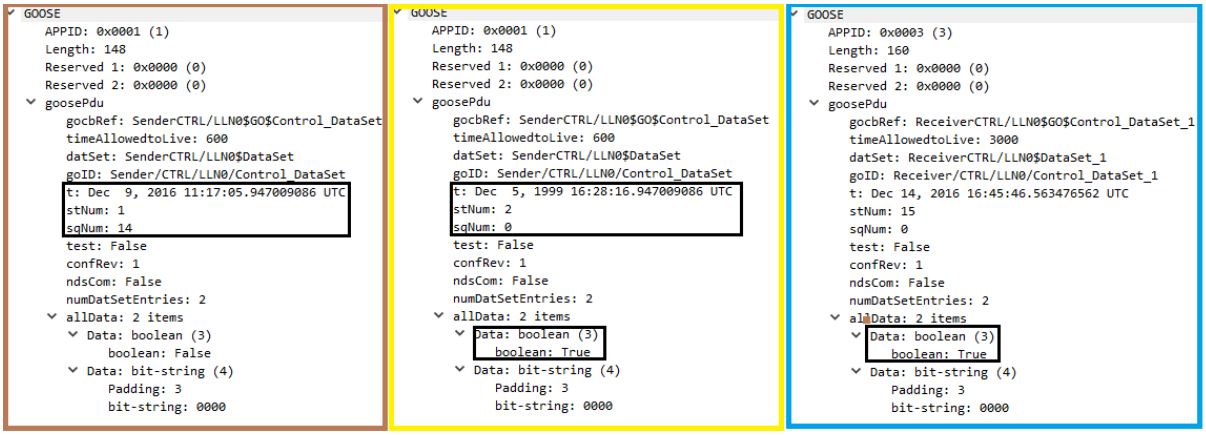


Figure 132 Incorrect Sequence – Time Sequence - Test two - GOOSEs detailed view

In conclusion, SIPROTEC 4 devices do not process the Time Stamp indication to define whether a GOOSE is valid or not.

Despite that, it has been demonstrated that IEC 61850 devices are able to detect an incorrect sequence of messages, as stated in IEC 61784-3, through the parameters State Number and Sequence Number.

During this last test, more than one parameter was modified in the additional GOOSE and all needed to be compliant with the expected ones. Speaking of safety, this represent a very unlikely scenario, but it has shown an important security issue that will be widely discussed in security analysis section

3.3.4.4 UNACCEPTABLE DELAY TEST

Unacceptable delays occur each time a message is delayed beyond its permitted arrival time window. In order to prove the ability of IEC 61850 to detect this kind of communication error, two tests were done.

During the first test, the GOOSE model with message fifteen delayed by 500ms was used (Figure 133)

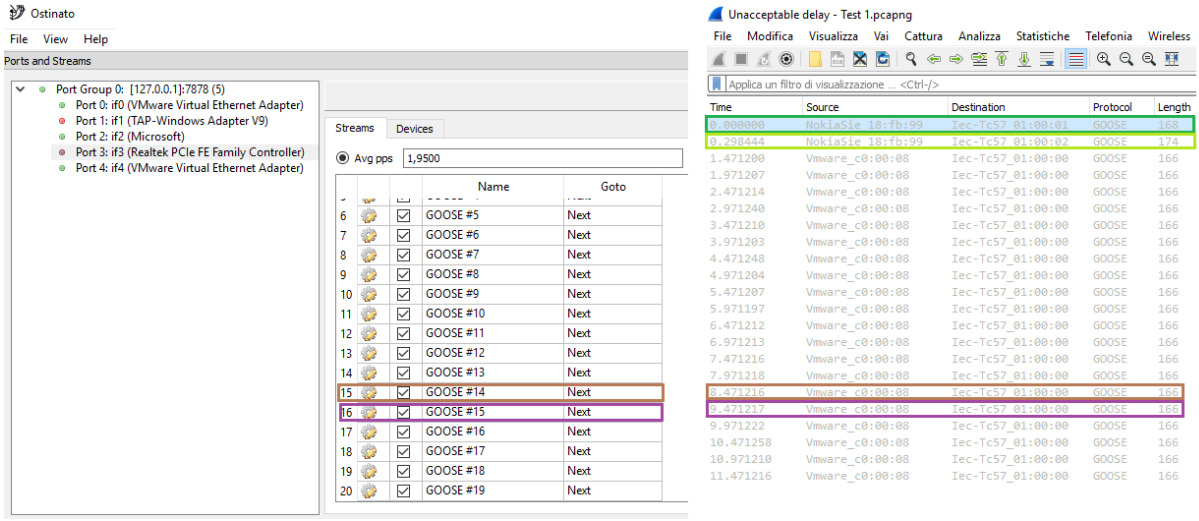


Figure 133 Unacceptable Delay - Test one – Ostinato's GUI on the left side and Wireshark capture on the right side

As GOOSE number fifteen arrived after 1000ms from the fourteenth, it arrived within its allowed time window, therefore it was classified as valid and acknowledged without further indications.

The second test was conducted with the same philosophy of the previous, but for GOOSE fifteen that was delayed by 1000ms (Figure 134). During this test, GOOSE fifteen arrived after 1500ms from the fourteenth's arrival and, since it arrived after its time window was elapsed, exactly after 1200ms a communication error was reported (red square).

When GOOSE number fifteen finally arrived (pink square) the Not Valid indication returned to FALSE and a new GOOSE was sent (turquoise square). This issue did not

affect the detection mechanism as, before GOOSE fifteen arrived, an alarm was already sent.

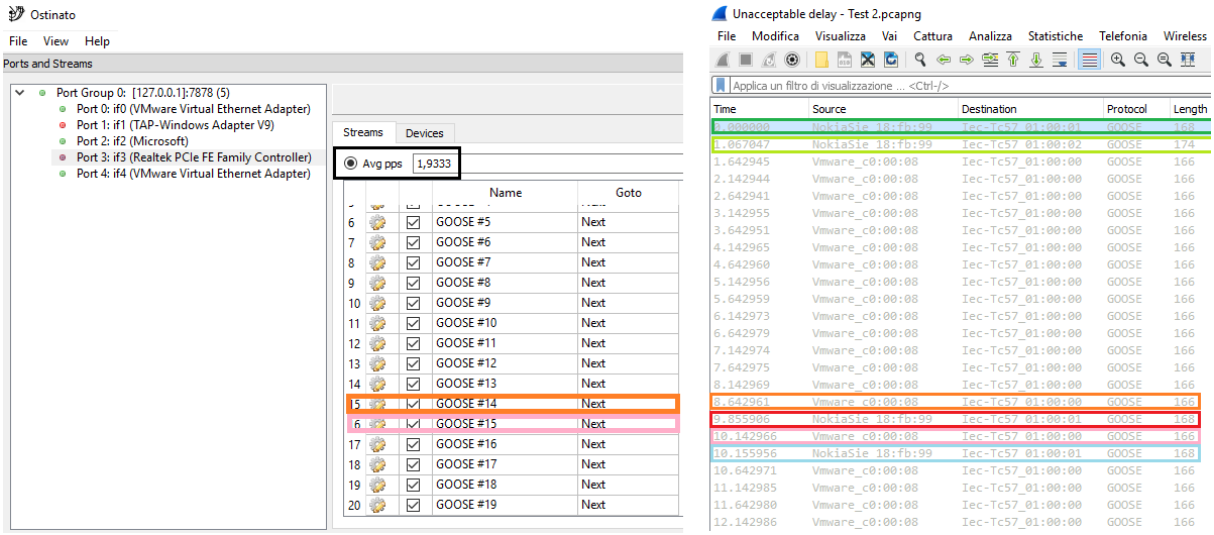


Figure 134 Unacceptable Delay - Test two – Ostinato's GUI on the left and Wireshark capture on the right

A detailed view of the affected GOOSEs is provided by Figure 135.

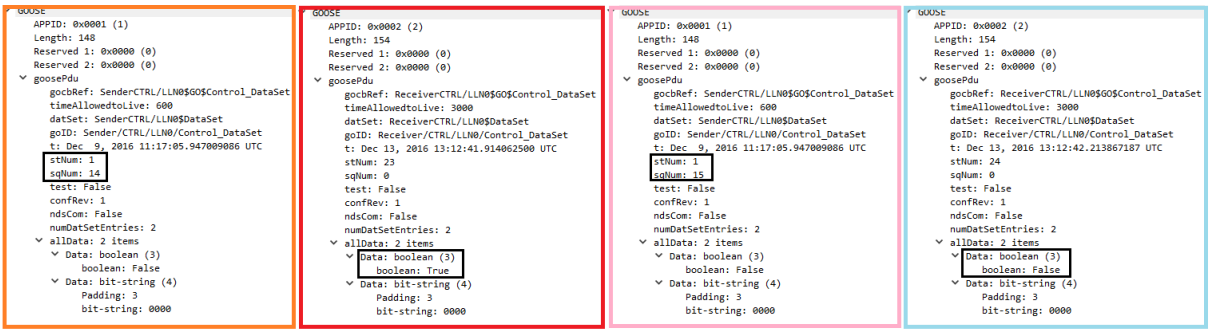


Figure 135 Unacceptable Delay - Test two - GOOSEs detailed view

This test has shown that, each time a message arrives beyond its allowed time window, the receiver is able to detect such an issue.

To conclude, IEC 61850 compliant devices can detect a communication error, due to unaccepted delay, as defined in IEC 61784-3.

3.3.4.5 UNINTENDED REPETITION

Unintended repetition occurs each time a message referring to an already acknowledged state is received at an incorrect point of time. Those messages are

additional to the normal data flux, carrying old or not-updated information, therefore they must be detected and the corresponding countermeasures have to be taken.

In order to test SIPROTEC's behaviour in case of unintended repetitions, two tests were done. Both of tests aim to determinate whether the repeated message is classified as valid or not.

The first test was done starting from the GOOSE model whose telegram ten is repeated between the fifteenth and the sixteenth message. The additional GOOSE does not affect the normal flux of data, so that the average speed results slightly higher than 2pps (Figure 136).

As shown in Figure 136, when the repeated message was received, the IED did not provide any alarm indication. This means that such a GOOSE was not acknowledged, otherwise an error GOOSE would have been sent after 1200ms.

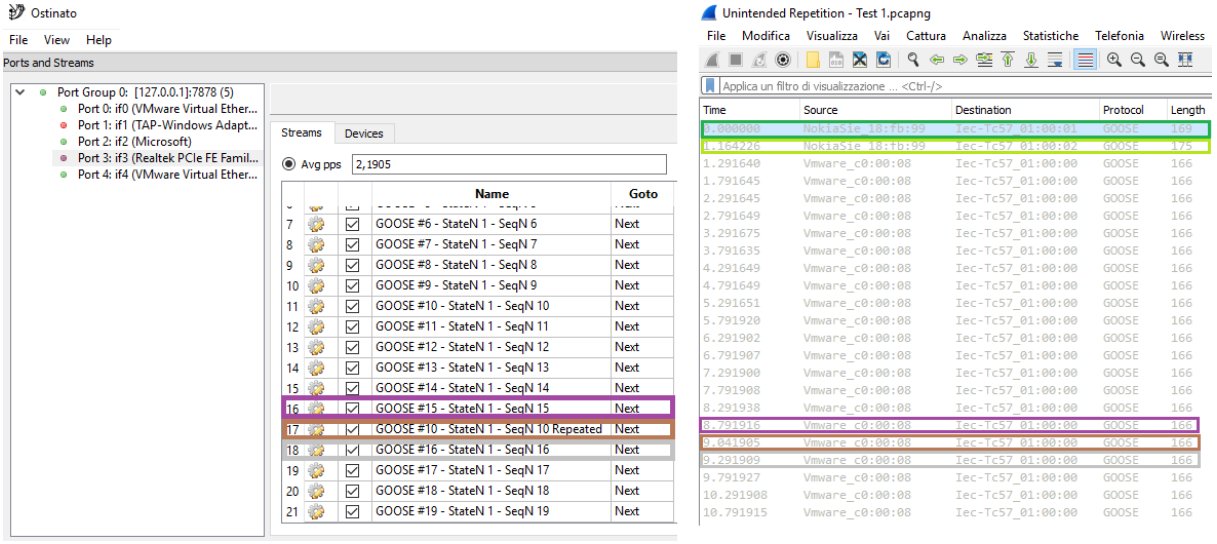


Figure 136 Incorrect Sequence - Test one – Ostinato's GUI on the left and Wireshark capture on the right

In order to confirm that such a message was ignored, in the second test, the repeated GOOSE carried also a different State Number, Time Stamp and Data value.

The second test was done by means of a different GOOSE model with one repeated message and, as in the previous test, the repeated message is additional to the normal

flux of data (Figure 137). To achieve this, the latter was put in between messages fifteen and sixteen (violet square).

A detailed view of the affected messages is provided by Figure 138.

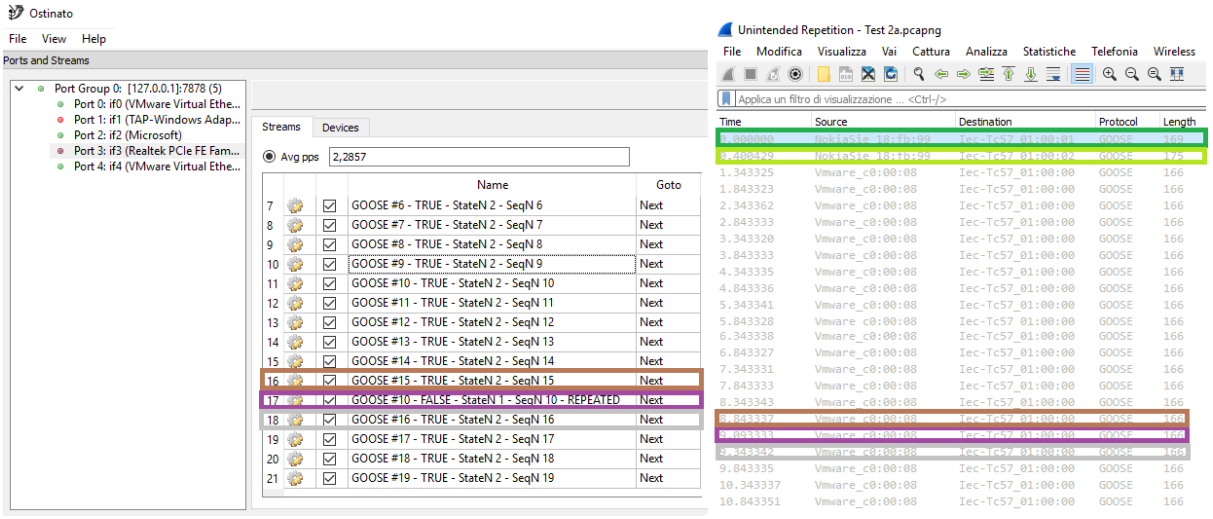


Figure 137 Unintended repetition - Test two - Ostinato GUI on the left and Wireshark capture on the right

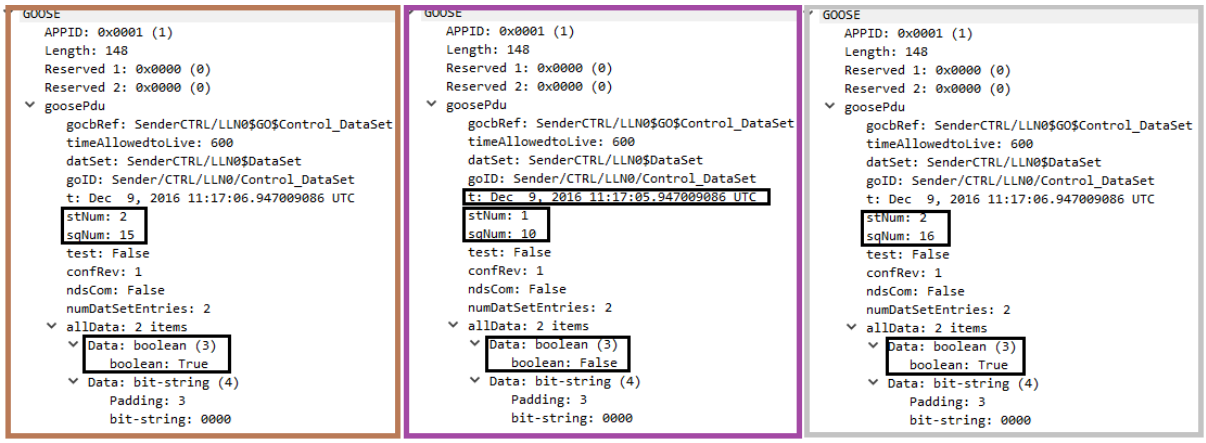


Figure 138 Incorrect Sequence - Test two - GOOSE detailed view

Below is a list of the changes done to the model:

- State Number: each message has such a parameter's value equal to 2, except for the eighteenth packet whose value is equal to 1 (violet square)
- Sequence Number: the repeated message has a Sequence Number value equal to 10 (violet square), all the other GOOSEs follow the usual order

- Time Stamp: this value is equal to Dec 9, 2016 11:17:06 UTC except for the eighteenth packet (violet square), which refers to a previous state, whose value is equal to Dec 9, 2016 11:17:05 UTC
- Data: this value is equal to TRUE except for the eighteenth packet (violet square), which refers to a previous state, whose value is equal to FALSE

This test showed that the communication error was detected, the repeated message was classified as invalid and, therefore, not acknowledged. In fact, if such a GOOSE was acknowledged, a response GOOSE would have been sent.

In conclusion, IEC 61850 can detect communication errors due to unintended repetitions, as defined in IEC 61784-3, by means of Sequence and/or State Numbers. Therefore, all the GOOSEs referring to an old not-updated value are classified as such and ignored.

3.3.4.6 ADDRESSING

Addressing occurs each time a safety relevant message is sent to the wrong safety relevant participant, which treat the reception as correct. In order to test the ability of IEC 61850 to detect such a communication error, two tests were performed.

The first test emulates the scenario in which all the safety relevant IED belong to a unique VLAN network and, in this case, for both sender and receiver, VLAN ID 2 was used.

In GOOSE communications, as telegrams are usually sent by multicast mechanism, the destination of each GOOSE is not defined at sender side. Each IED, depending on the way it was programmed, decides whether to subscribe a GOOSE message or not. If a GOOSE is received by an IED which was not programmed to subscribe that message, then the latter is ignored.

In order to emulate an addressing error, the GOOSE model was modified with two additional messages, each of which contains the following altered parameters:

- Source MAC address: in order to identify another source, a value of 00:5a:56:c0:00:08 was used
- State Number: this field was made equal to 2 for both additional GOOSEs
- Sequence Number: for the additional messages, this parameter was made respectively equal to 0 and 1
- Data: this field was made equal to TRUE for both additional GOOSEs
- gocbRef: this field must be compliant with the GOOSE application used, for this purpose, SenderCTRL/LLN0GOControl_AddTest was chosen
- dataSet: this field must be compliant with the GOOSE application used, for this purpose, SenderCTRL/LLN0$AddTest was chosen
- goID: this field must be compliant with the GOOSE application used, for this purpose, Sender/CTRL/LLN0/Addressing_test was chosen

During this test, it was assumed that each safety function has its own goID, gocbRef and dataSet parameters. In general, a message with unexpected State or Sequence Numbers is ignored regardless of the other parameter's values.

For example, when a safety-related participant sends a GOOSE to the wrong recipient, if the latter does not expect a message with exactly those State or/and Sequence Numbers, then that message would be classified as invalid regardless of goID, gocbRef and dataSet parameters.

In this test, State Number and Sequence number were chosen to make the incoming GOOSE available for subscription, although it is very unlikely that a message with the exact needed parameters is sent just to address an error.

The last three parameters identify the GOOSE application name, each IED reads these fields to decide whether it can subscribe to that telegram or not. For this purpose, the receiver was not programmed to subscribe to GOOSEs with such parameters, since they refer to another GOOSE application.

These two GOOSEs (yellow and grey squares) were inserted additionally to the normal data flux, so that the average sending speed resulted slightly greater than 2pps (Figure 139).

If one of the two additional messages was acknowledged, a response GOOSE would have been sent by the IED.

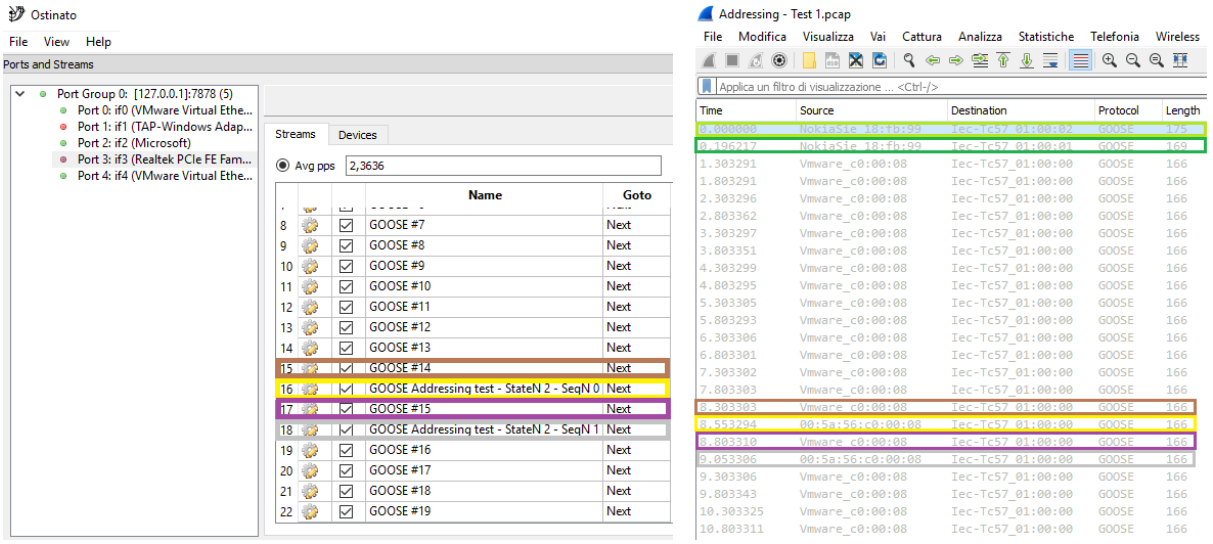


Figure 139 Addressing Test - Test one – Ostinato's GUI on the left and Wireshark capture on the right

A detailed view of the affected telegrams is provided by Figure 140.

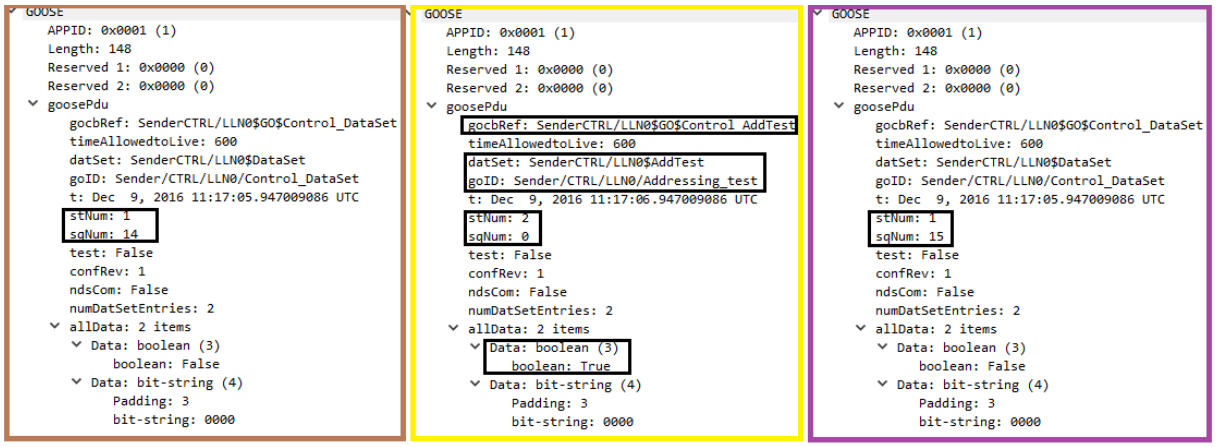


Figure 140 Addressing Test - Test one - GOOSEs detailed view

Since no response GOOSE was captured, it is confirmed that all the GOOSEs originating from a wrong safety relevant participant are detected and thus not processed.

The second test represents a scenario in which the two safety relevant participants belong to two different VLAN networks. For this purpose, for additional GOOSEs, VLAN ID 4 was used.

This test follows exactly the same methods as the previous one, except for VLAN TAGs, this time the Sender and the Receiver belong to different logical networks. Wireshark captures both on sender side and on receiver side are shown in Figure 141.

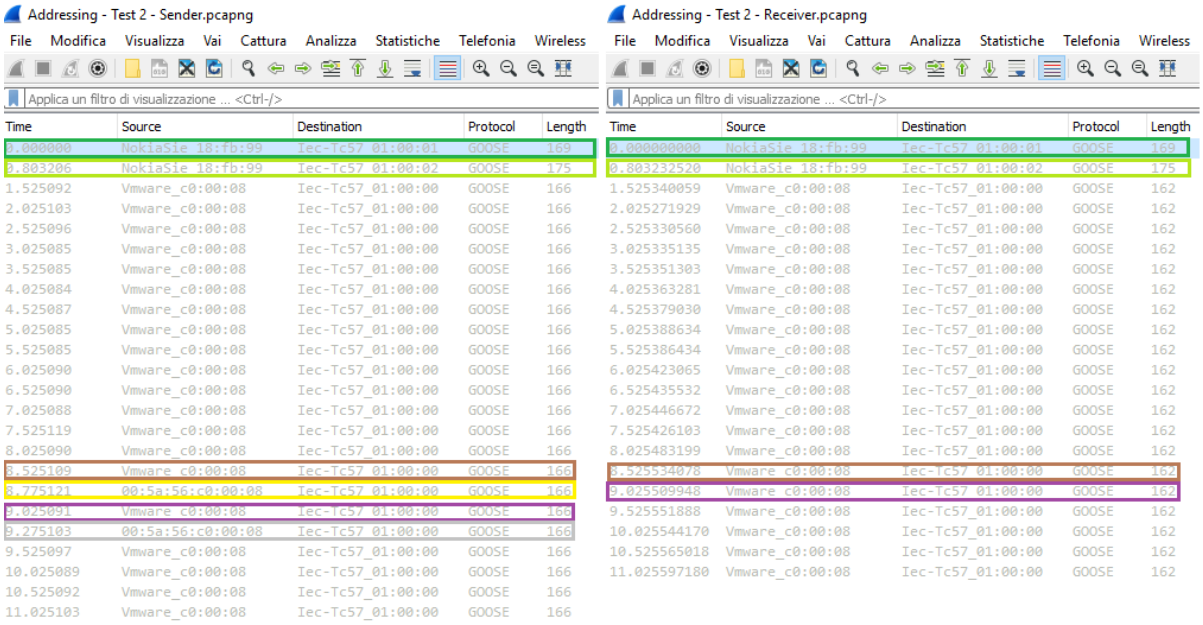


Figure 141 Addressing - Test two - Wireshark captures - Sender side on the left and receiver side on the right

In this case, none of the additional messages were received as they were filtered by the switch.

To conclude, IEC 61850 can detect the addressing communication error as defined in IEC 61784-3. If the affected safety relevant participants belong to different logical networks, then the error is avoided directly by the switch. If the affected safety relevant participants belong to the same network, the receiver detects that the incoming message is part of those the IED can subscribe to, therefore the GOOSE is ignored.

3.3.4.7 INSERTION

Insertion communication problem occurs each time a message related to an unexpected/unknown source identity is inserted in the communication.

Each publisher sends GOOSEs with a predefined set of parameters belonging to those specific messages, for example Sequence Number, State Number and GOOSE ID. Through these parameters, an IED which receives a certain number of messages can recognize and acknowledge only those messages to which it is programmed to subscribe. All other received messages are ignored so that, even if an insertion error occurs, that specific inserted message can be detected and ignored.

Furthermore, it is possible to prevent mitigate insertion errors by setting up a VLAN network only for safety related IEDs, so that no unexpected or unknown source identity can send a message within that network.

In order to prove that insertion communication errors can be detected, two tests were done.

The first test was done without using different VLANs, this means that both the SIPROTEC and PC belong to the same VLAN network, in this case VLAN ID 2 was used. In order to perform an insertion test, the GOOSE model was modified with two additional messages (yellow and brown squares in Figure 142).

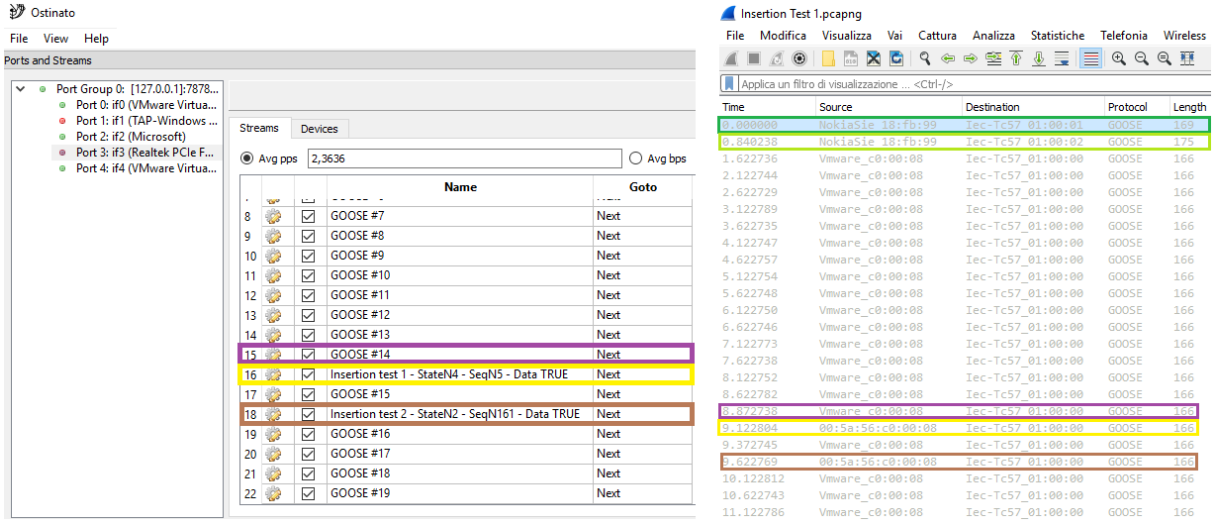


Figure 142 Insertion - Test one – Ostinato's GUI on the left and Wireshark capture on the right

The first inserted GOOSE was modified as follows:

- Source MAC Address: its value was set equal to 00:5a:56:0c:00:98
- Time Stamp: Time indication was set equal to Dec 9, 2016 11:17:06 UTC
- State Number: its value was set equal to 4
- Sequence Number: its value was set equal to 5
- Data: the data value was set equal to TRUE.

The second inserted GOOSE were modified as follows:

- Source MAC Address: its value was set equal to 00:5a:56:0c:00:98
- Time Stamp: Time indication was set equal to Dec 9, 2016 11:17:06 UTC
- State Number: its value was set equal to 2
- Sequence Number: its value was set equal to 161
- Data: the data value was set equal to TRUE.

In order to subscribe the inserted GOOSEs, the parameters GoID, gocbRef and dataSet parameters were not modified. That said, section 3.3.4.6 showed that, if a message with such altered parameters is received, the IED ignores that GOOSE since it does not belong to the GOOSEs the device is supposed to subscribe.

As shows Figure 142, at reception of both the inserted GOOSEs no response GOOSE was sent, this means that the IEC 61850 device recognized those messages as being from a communication error and, therefore, they were ignored.

The second test follows the same methodologies of the previous one; it was done using VLAN TAGGING feature so that, while ordinary GOOSEs were sent with VLAN ID equal to 2, the additional GOOSEs were sent with VLAN ID equal to 5.

Figure 143 shows Ostinato’s configuration in order to send different GOOSEs with different VLAN TAGs.

In order to capture both sent and received packets, two Wireshark instances were used: one on the sending side2 and another within the VLAN 2 (Figure 144).

Assuming that VLAN ID 2 is reserved for safety related devices only, so that no unknown source can access to this network, even if an insertion error occurs, the

inserted message cannot reach any of the safety related devices (receiver side in Figure 144).

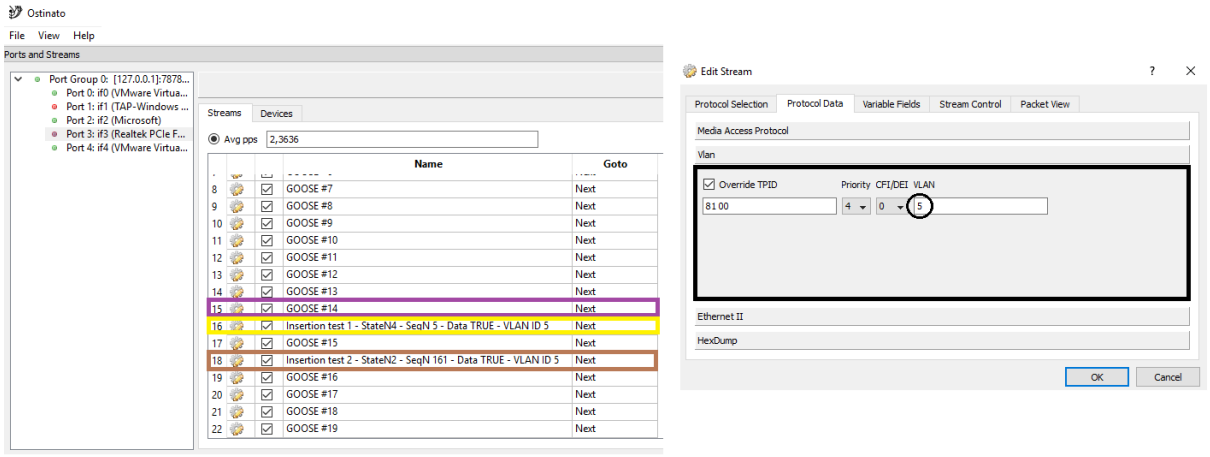


Figure 143 Insertion - Test two – Ostinato's GUI on the left and VLAN setup on the right

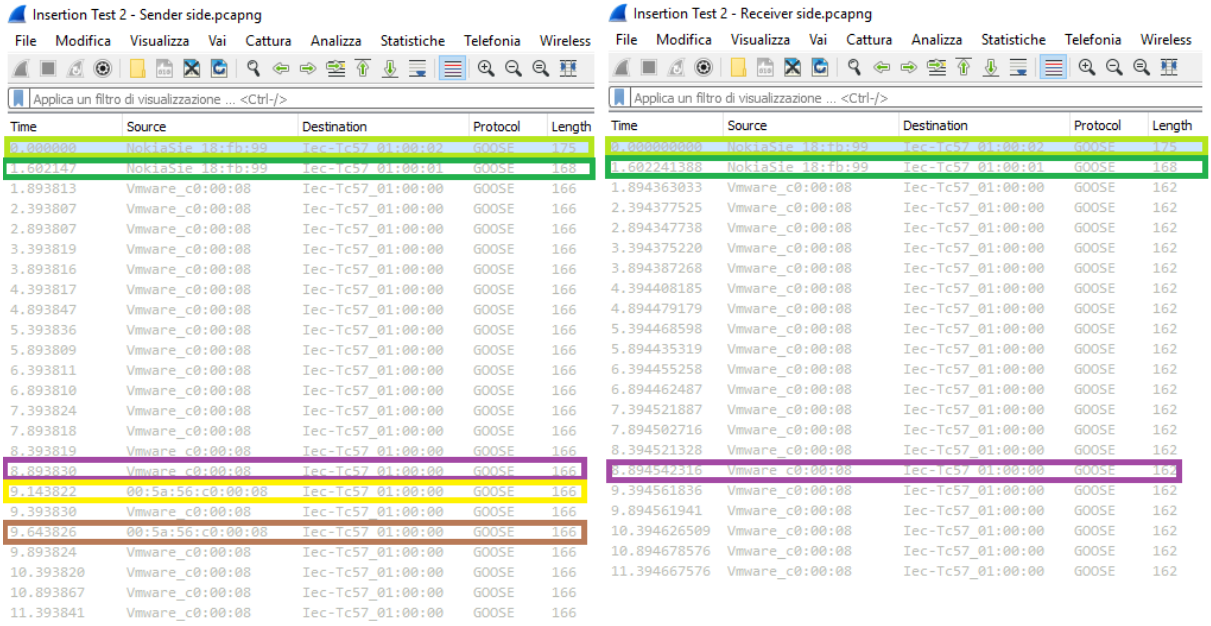


Figure 144 Insertion - Test two - Wireshark captures, sender side on the left and receiver side on the right

To conclude, IEC 61850 can detect the insertion communication error as defined in IEC 61784-3, therefore it does not affect the regular communication. Additionally, such a communication error can be mitigated by means of IEEE802.1Q use. In fact, if a VLAN network is reserved only for safety related purposes, no message from unknown/unexpected sources can be inserted within that network.

3.3.4.8 MASQUERADE

Masquerade communication error occurs each time a message that relates to an apparently valid source entity is inserted, so a safety related participant, which then treats it as safety relevant, may receive a non-safety relevant message.

In order to establish the reaction to a masquerade error, two tests were done. The first test was conducted without the use of the IEEE 802.1Q feature and the second one using two different VLAN networks.

During the first test, one additional message was inserted between GOOSEs fifteen and sixteen.

As seen in the previous sections, if an incoming message does not belong to the one the IED is set up to subscribe to, regardless of all the GOOSE parameters, that message is ignored by the receiver. To prevent this, during both of the following tests, the parameters goID, gocbRef and dataSet of the additional message were left untouched. Plus, to simulate a valid source, the source MAC address of the affected message was left untouched as well.

For this test, the hypothesis that all the IEDs belong to the same VLAN network was assumed.

The additional GOOSE was modified as follows:

- Time Stamp: Time indication was set equal to Dec 9, 2016 11:17:06 UTC
- State Number: its value was set equal to 2
- Sequence Number: its value was set equal to 0
- Data: the data value was set equal to TRUE

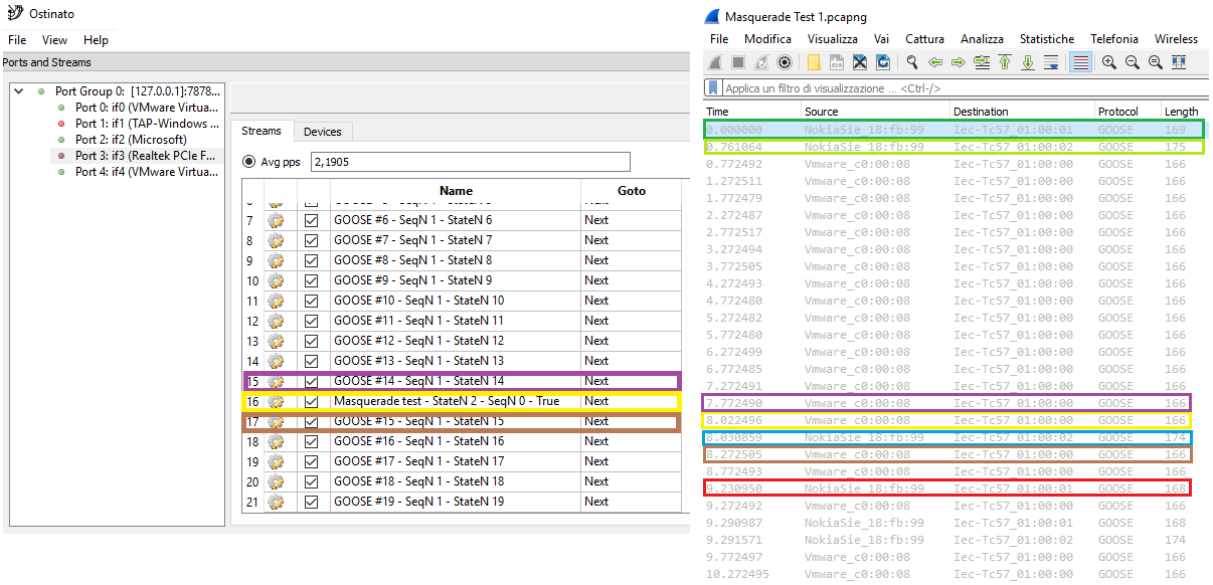


Figure 145 Masquerade - Test one – Ostinato's GUI on the left and Wireshark capture on the right

As Figure 145 shows, the inserted message (yellow square) was received and acknowledged as a valid GOOSE, starting from this point, the IED waited a time equal to 2*TAL before sending an alarm GOOSE (red square). In this case, the alarm was sent because of the wrong reason, in fact, the additional message was subscribed as a valid GOOSE and the masquerade error was not detected in the first place.

Since the source identity is not checked, if the affected GOOSE is part of the GOOSEs the protection is programmed to subscribe, then IEC 61850 is not able to detect these kind of masquerade errors without the VLAN usage.

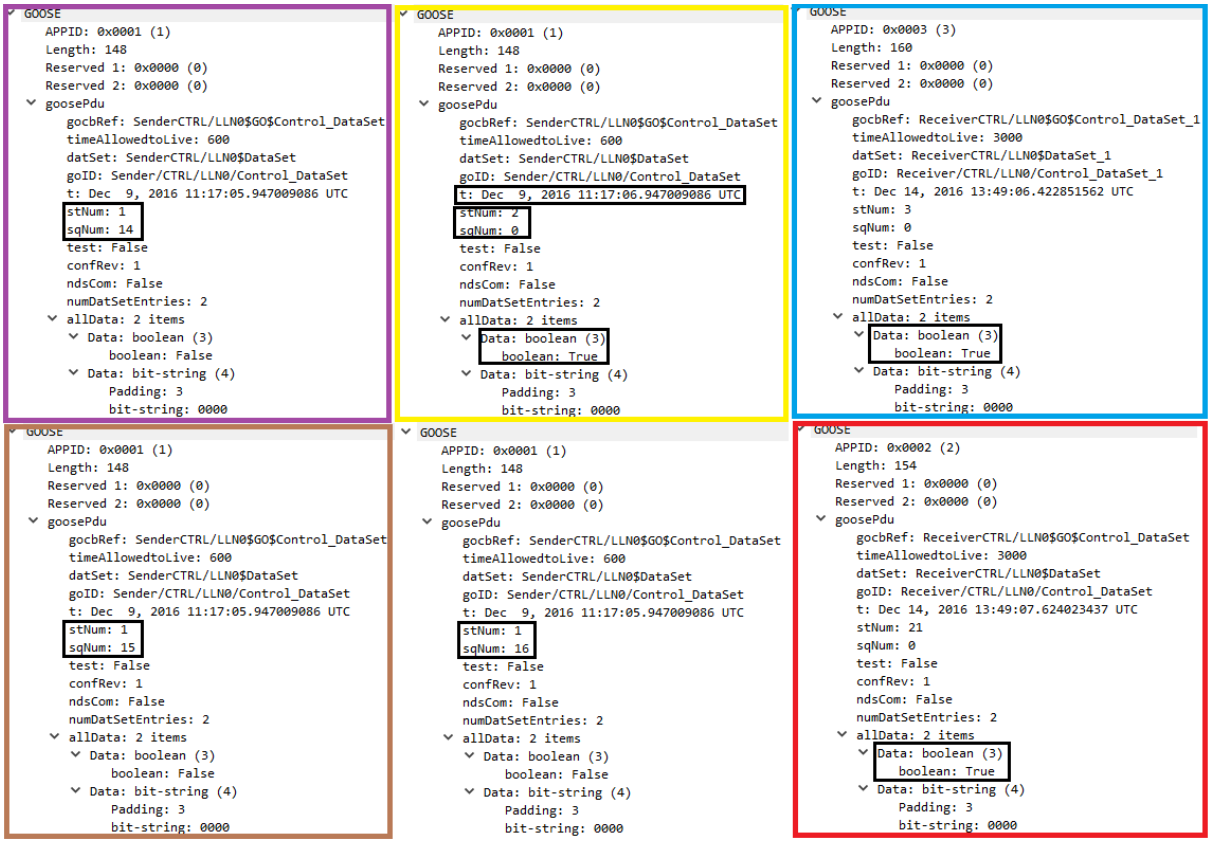


Figure 146 Masquerade - Test one - GOOSEs detailed view

The second test was done using IEEE 802.1Q feature, so that for all the non-safety relevant devices, VLAN ID 5 was used. The usage of VLAN TAGGING prevents the communication between a non-safety relevant IED and a safety relevant one.

As shown in Figure 147, this test follows the same mechanism of the previous one but using IEEE 802.1Q feature. This time the additional GOOSE was inserted between telegrams fourteen and fifteen. As expected, the test GOOSE did not pass through the switch, so that on receiver side, no masquerade error was detected (Receiver side in Figure 148).

This kind of configuration can prevent any type of masquerade error, also those errors whose GOOSE parameters are compliant the ones the IED can subscribe.

A detailed view of the affected GOOSEs is provided by Figure 149.

To conclude, the IEDs compliant with IEC 61850 can detect/prevent all the masquerade communication errors, as defined in IEC 61784-3, using the IEEE802.1Q feature.

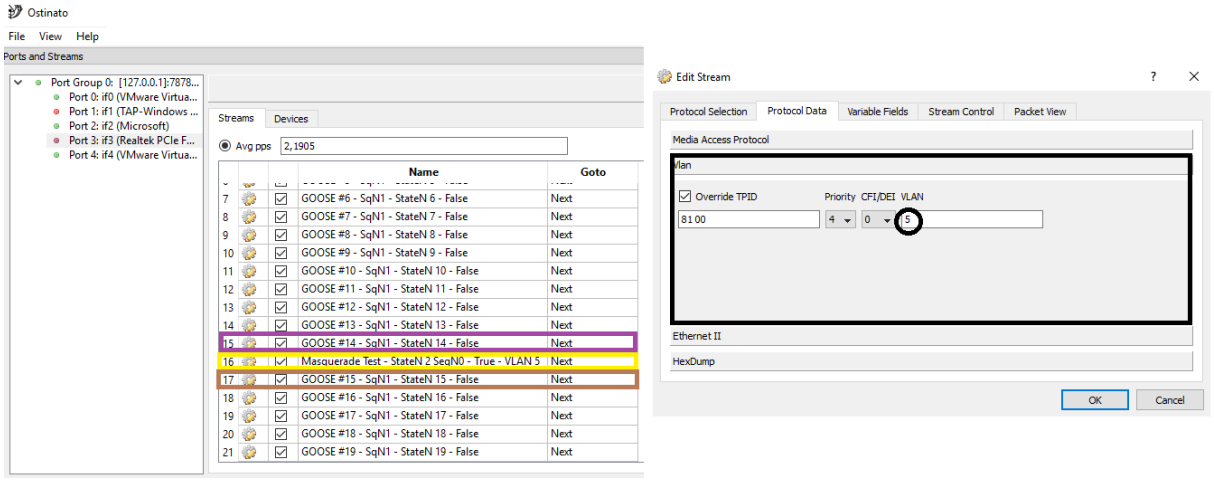


Figure 147 Masquerade - Test two – Ostinato's GUI on the left and VLAN setup on the right

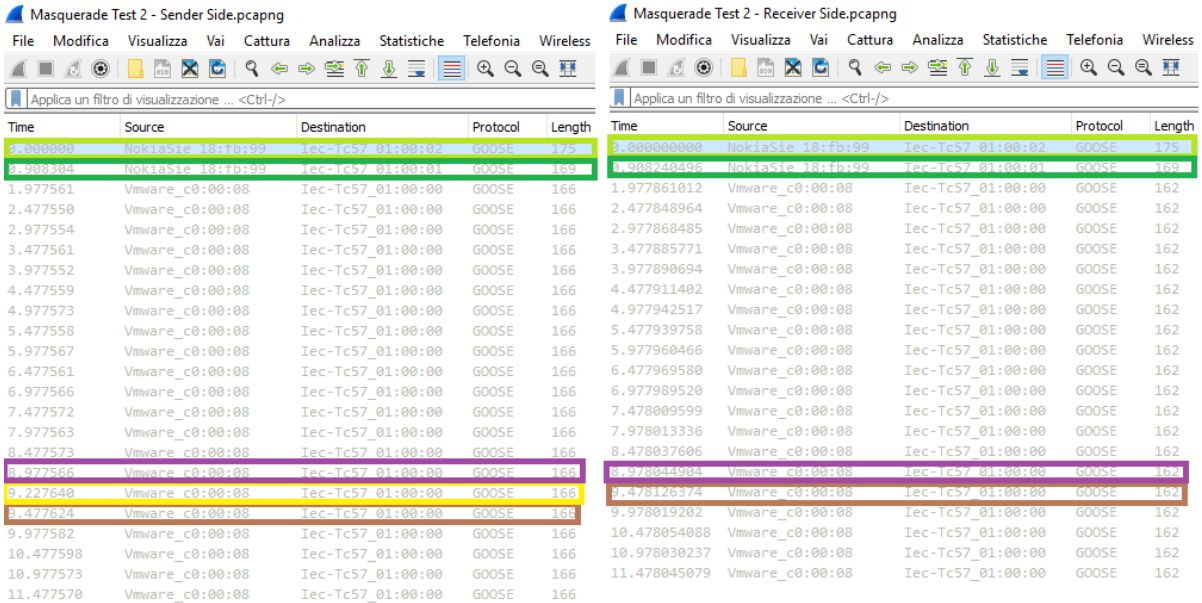


Figure 148 Masquerade - Test two - Wireshark captures, sender side on the left and receiver side on the right

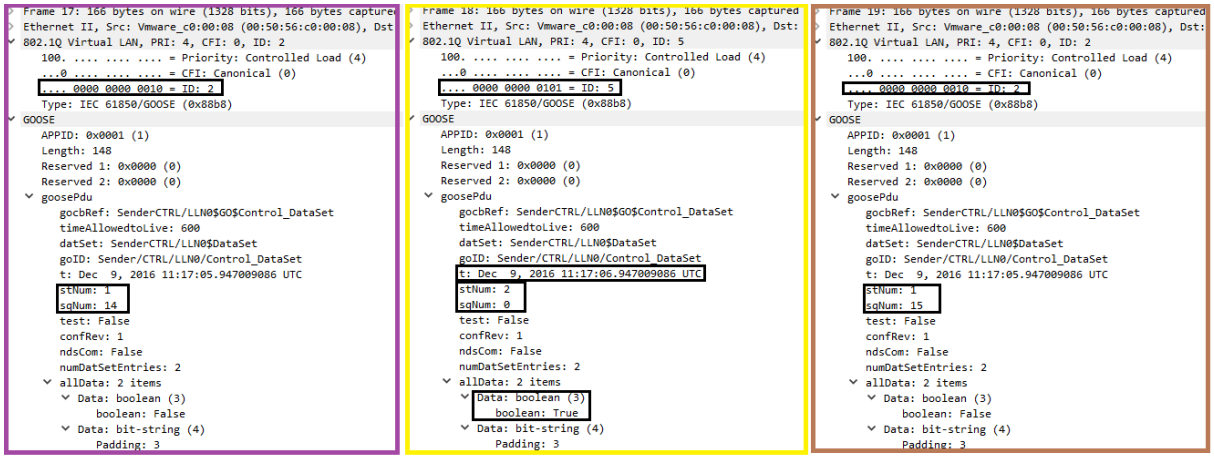


Figure 149 Masquerade - Test two - GOOSEs detailed view

3.3.4.9 CORRUPTION

Messages may be corrupted due to errors within a bus participant, errors on the transmission medium, or message interference.

IEC 61850 uses the Ethernet as layer two of the ISO/OSI model, so that it takes advantages of its FCS feature for corruption communication error.

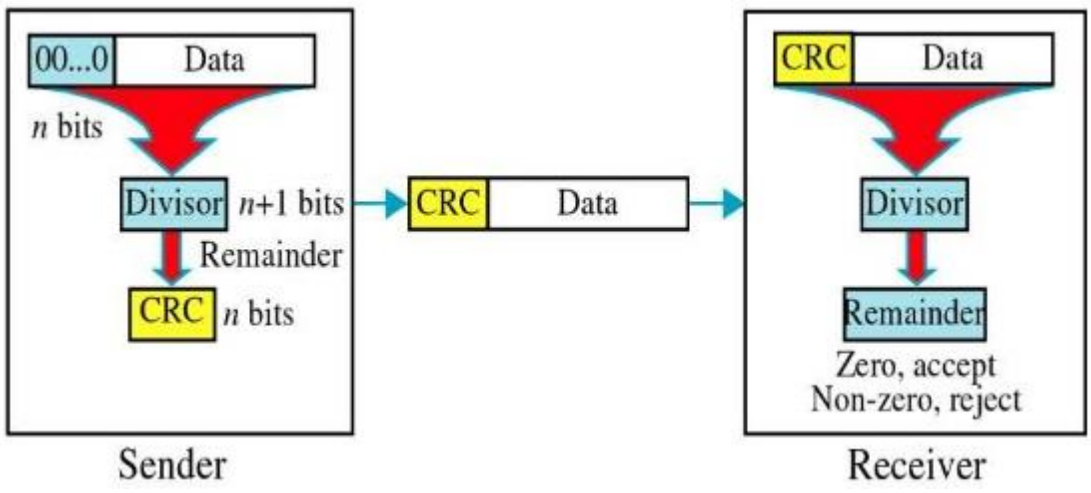


Figure 150 Ethernet CRC

The FCS frame is managed directly by the network card and it is not accessible through the software used in this test. Each time an Ethernet frame is sent, the network card calculates the proper CRC-32 and the latter is then added at the end of the Ethernet frame. On the other hand, when an Ethernet packet is received, the network card

checks whether the frame is corrupted or not through the FCS. If the CRC test is passed, the FCS is removed and the frame is forwarded to the upper layer.

For these reasons, it is hard to access to the FCS feature with conventional software. Furthermore, it is a specific characteristic of the Ethernet protocol and not of IEC 61850.

Additionally, IEC 61850 can implement some security characteristics as stated in IEC 62351-6, in fact the Reserved 2 field of the GOOSE can provide a further 16-bit CRC.

Nowadays there is no IEC 61850 IED which implements IEC 62351-6 Security GOOSE feature but, soon, it will provide an additional hash function regardless of the one used by the communication layer. Theoretically, IEC 61850 can perform a double integrity check, the first at link layer (level two) and another at application layer (level seven) of the ISO/OSI model.

Furthermore IEC 61850 uses a full-duplex communication by means of Ethernet switches with a node-to-switch or switch-to-switch configuration, such that the network can be considered completely collision-free.

For the reasons stated above the proper functioning of the Ethernet checksum mechanism has been assumed without further test.

3.3.4.10 CONCLUSION

As shown in the previous chapters, each one of the communication errors listed in IEC 61784-3 was replicated by means of the software/hardware configuration stated in 3.3.1. The tests were performed with an IEC 61850 compliant device which, during each of the performed tests, detected the corresponding communication error.

During these tests, each time a communication error was detected, the SIPROTEC was programmed to send an alarm GOOSE. IEC 61784-3 does not define what must be done when a communication error is detected. It is reasonable to think that, for electrical plants, the safe state is considered a circuit de-energized. A possible solution is to configure the IED to switch the system in a safe mode when a communication

error is detected using the IED's configuration tool. Thus, in an actual context, the IED can be programmed to switch the system in a safe condition through its output relays, instead of sending a GOOSE.

In this case, as the configuration software is used to program one or more safety functions in the device, the software itself should be certified as well.

Concluding, IEC 61850 has proved to be fully compliant with all the IEC 61784-3 requirements. Thus, both hardware and software part shown previously may be certificated in order to implement safety-related functions by using devices compliant with IEC 61850.

Table 5 IEC 61850 safety countermeasures summary

Communication errors	Safety measures			
	Sequence number [StNum + SqNum]	Time expectation [timeAllowedtoLive]	Connection authentication [IEEE 802.1 Q]	Data integrity assurance [FCS]
Loss	x	x		
Incorrect sequence	x	x		
Unacceptable delay		x		
Unintended repetition	x			
Addressing			x	
Insertion	x		x	
Masquerade			x	
Corruption				x

4. SECURITY ANALYSIS

4.1 CYBER SECURITY OVERVIEW

Security issues for the power industry have become increasingly relevant during the past decade as the industry has relied more and more on communication protocols. In this chapter, a practical attack by exploiting weaknesses in GOOSE is demonstrated. The security issue becomes important each time an attack like this can have devastating consequences on the reliability of the grid and it can create a widespread interruption in power generation and distribution[27][29].

This chapter aims to show how to an attacker can capture, alter, and re-inject GOOSE messages into the network. By taking advantage of existing security holes in the GOOSE messaging protocol, it will be shown as the holes could be used to significantly disrupt the power grid and highlights the need to apply security measures in this area.

In the early days of the IEC 61850 there were no recommendations for security on the layer 2 multicast GOOSE and SMV messages. The vulnerability was considered low because the messages were running in a confined network inside a substation protected by the physical network isolation. This is not true today when new applications are running GOOSE messages outside substations for wide-area transmission protection schemes and distribution automation schemes. Further, substations have become more connected to external networks and employ wireless networks with the potential to expose their IEC 61850 network to outside attackers.

An attack is defined by the motivation, vectors, and the techniques. Below a motivated attacker is assumed and the attention is focused on the attack vectors and techniques. The attack vector is a path or means by which an attacker gains access to a computer or network in order to achieve their ultimate goal. There are several layer 2 attack techniques that could be applied to GOOSE message since the underlying IEC 61850 network is Ethernet. An Ethernet attack could be created using a variety of techniques, and the structure of protocols in the OSI model are such that the upper layers in the model could be unaware that layer 2 has been compromised.

There are several consequences if a layer 2 attack is executed in a substation. The main purpose of the GOOSE message is to carry vital information (alarms, status, and control) between devices. Any alteration of these values could create an automation breakdown, causing a circuit breaker to miss an operation, bypassing interlocks, or causing physical damages in the field devices like power transformers or circuit breakers. If the attack compromises a bus bar or differential protection, more than one distribution or transmission circuit could be affected. As a result, one part of the city or region would suffer an outage. If the same attack involved transmission or generation circuits, the outage could trigger cascading failures and become sufficiently large to affect complete cities or states.

In order to perform the security test, a GOOSE exploit via spoofing was used. The latter consists of an intruder which publishes false layer 2 packets and a device on the receiving side that mistakenly believe it is receiving valid packets sent by a trusted or secured entity. This attack is possible due to the unencrypted/unauthenticated nature of GOOSE messages.

4.2 IEC 62351 AND SECURE COMMUNICATIONS

IEC 62351 standard arises from the needs to add security features in the protocol series drawn up by the TC57. IEC 61850 has gained global acceptance by both vendors as well as customers. Cyber security on the other hand has quickly become one of the most dominant topics for control systems in general and electrical utilities. The combination of the two, securing IEC 61850 based communications, has been one of the goals of the recently published technical specification IEC 62351[26].

This chapter will highlight the challenge of addressing secure communication in the substation real-time environment, complying with the IEC 61850 real-time specifications. The major difficulties are to reach the real-time performance defined in IEC 61850 for GOOSE and SV data.

The scope of the IEC 62351 series is information security for power system control operations. Its primary objective is to undertake the development of standards for security of the communication protocols defined by IEC TC 57, specifically the IEC

60870-5 series, the IEC 60870-6 series, the IEC 61850 series, the IEC 61970 series, and the IEC 61968 series. IEC 62351 standard is currently divided into 8 parts as shown below.

Part	Title
1	Communication network and system security – Introduction to security issues
2	Glossary of terms
3	Security for profiles including TCP/IP
4	Profiles including MMS
5	Security for IEC 60870-5 and derivatives
6	Security for IEC 61850
7	Network and system management (NSM) data object models
8	Role-Based Access Control
	Key Management (Certificate Handling)
	Security Architecture

Figure 151 Contents of IEC 62351

As discussed in the previous chapters, IEC 61850 communications can be divided into client server (i.e. MMS) and real time (i.e. GOOSE and Sample Values) communications. IEC 62351 provides different methods for securing the different communication types:

- MMS (IEC 61850-8-1): securing MMS traffic is done on the application and the transport level. Peer authentication is performed on the application level. Authentication information comprises a X.509 encoded certificate, a time stamp and the digitally signed time value. For security on the transport layer IEC 62351 refers to TLS. It also specifies a set of mandatory and recommended cipher suites to be used, at a minimum TLS_DH_DSS_WITH_AES_256_SHA1 and TLS_DH_RSA_WITH_AES_128_ SHA2 must be supported.
- GOOSE / Sampled Values: security of real-time traffic is limited to message authentication, i.e. use encryption is not specified. Message authentication is defined by extending the GOOSE / SV PDUs with an authentication value that is calculated by signing a SHA256 hash using RSA. Certificate exchange is not done as part of the messages; X.509 encoded certificates must be pre-installed on the receiving nodes.

Performance impacts should always be considered for any communication infrastructure before introducing encryption and / or message authentication. This is particularly true if asymmetric cryptography, real-time traffic or systems with limited resources are involved. In case of securing GOOSE and SV using IEC62315 all three constraints apply:

- Embedded devices such as Protection & Control IEDs or RTUs typically have little computational power (as compared to personal computers or servers) and only a (small) portion can be made available to functionality other than protection and control. In addition, changing or upgrading hardware is not an easy task for embedded devices that potentially have a very long lifetime. Security solutions for embedded devices should therefore not require major hardware changes.
- For both GOOSE and SV strict real-time constraints exist – 3ms response time for GOOSE and sampling rates up to 12 kHz for Sampled Values.
- IEC 62351, as of today, specifies the use of digital signatures (asymmetric cryptography using RSA) to authenticate broadcast GOOSE and SV packets.

Latency is one of the primary barriers to implement security for peer-to-peer communications between IEDs. For instance, IEC 61850-5 specifies a 4ms maximum delay for class P1 type 1A GOOSE messages related to breaker trip functions. As a result, encryption or other security measures, which increase the delay or latency, are avoided.

The IEC 62351 standard defines a mechanism that requires low computational power to authenticate the data adding a digital signature. The digital signature is created via mathematical techniques to validate the authenticity of a digital message using asymmetrical cryptography. This kind of scheme uses public and private keys to authenticate the message. The public key is shared with everyone to decrypt a hash of the message, while the private key is kept private by the publisher to sign the message. In the IEC 62351 standard part 6 states “for applications using GOOSE and IEC 61850-9-2 and requiring 4ms response times, multicast configurations and low

CPU overhead, encryption is not recommended". Nevertheless, the standard does not say anything about authentication and its limitation.

At present it is difficult to reconcile the needs for security and low latency. One study conducted by Cambridge University and ABB in 2010 showed that processing (encoding and decoding) digital signatures required intense CPU consumption. Therefore, 32-bit Intel and ARM cores are generally incapable of computing and verifying a digital signature using the Rivest, Shamir and Adleman (RSA) algorithm with 1024-bit keys within 4ms. The central processor unit (CPU) embedded in the IEDs has some restriction due to the power dissipation. The IEDs are fan-less; installed commonly in closed cases to avoid environmental issues like dust, water, or insects. New technologies like multiple cores may enable faster times within the same heat dissipation budget. However, there are many IEDs already installed in the market with slower CPUs.

Keeping in in mind the GOOSE message analysis made in 2.4.8, the following information constitute an extension to such messages. In fact, this section will show how some of the GOOSE frames can be used to add some security features to GOOSE message communication.

During the previous analysis, the presence of the Reserved 1 and Reserved 2 fields have been mentioned within the GOOSE structure and it is through these parameters that it is possible to address the security issue.

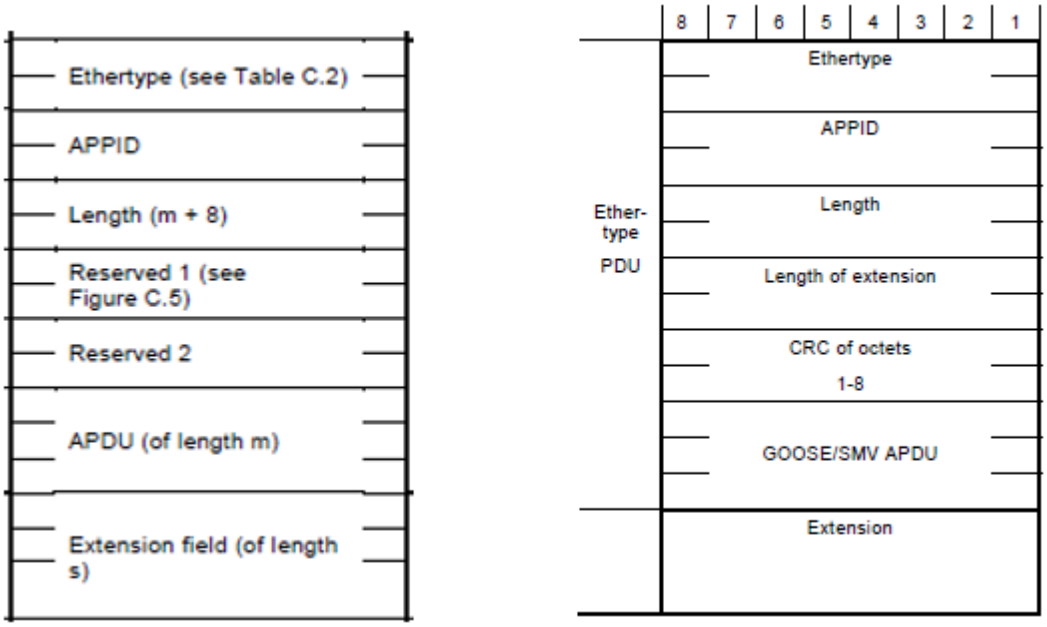


Figure 152 Normal GOOSE structure on the left, GOOSE structure with IEC62351-6 features on the right

The GOOSE structure in shown in Figure 152, particularly, on the left the structure of an un-protected GOOSE is shown, while on the right the structure of the Extended PDU, in accordance with IEC 62351-6, is shown.

Below is a list of the specification stated in IEC 62351-6 for the GOOSE structure:

- Reserved 1 field is used to specify the number of octets conveyed by the extension octets. This value is contained in the first octet of the Reserved 1 field. The valid range of values is 0 through 255, where a value of 0 indicate that no extension octets are present. The second octet of the Reserved 1 field is reserved for future use.
- Reserved 2 field contains a 16-bit CRC, as calculated per ISO/IEC 13239 (ISO HDLC). The CRC is calculated over Octets 1-8 of the VLAN information of the Extended PDU. The CRC is present each time the Extension Length has a non-zero value.
- Extension octets is an optional field and its length is defined on the Reserved 1. It contains the following information:
    - The Reserved Sequence is used to reserve future standardized extension for this specification. If no extension, besides Authentication

and Encryption is defined in this specification, this sequence is not be present.

- The Private Sequence is provided to allow vendors to convey Private information. The scope of the semantics and syntax of the contents of this sequence is out-of-scope of this specification and shall only be interoperable via prior agreement. This SEQUENCE is present only if there are actual contents being conveyed.
- The AuthenticationValue is based upon the generation of a reproducible Message Authentication Code (MAC). The MAC is generated through the computation of a SHA256 hash per RFC 4634. The hash contains all octets of the Extended PDU except for the Tag, Length and Value of the AuthenticationValue. The value of the hash is then digitally signed. Additionally, implementations that use the AuthenticationValue shall provide a public X.509 certificate for installation on the receiving clients.

The arrangements explained above are used in order to introduce the predefined security features and, in this regard, Figure 153 shows the differences between the two structures.

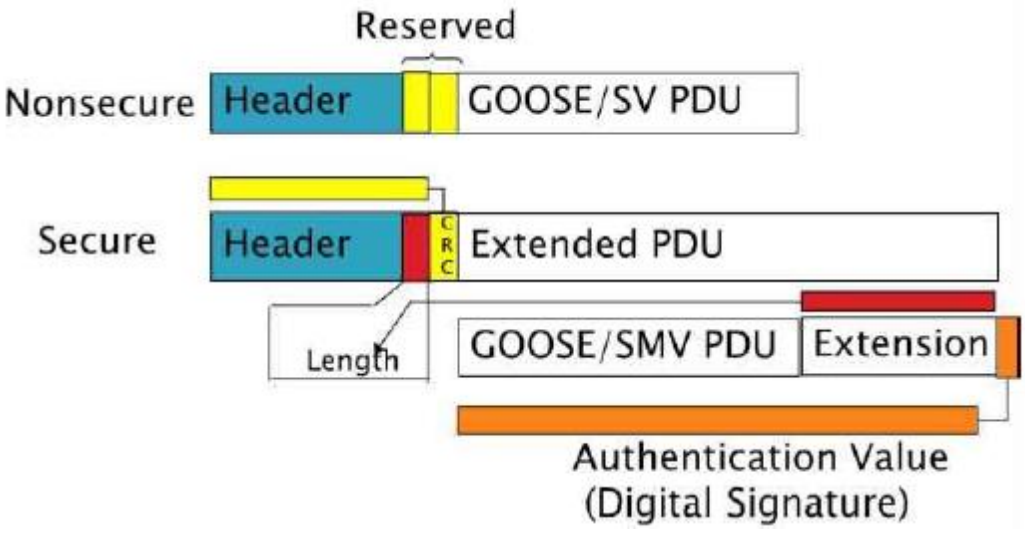


Figure 153 Differences between the two GOOSE structures

In order to achieve a secure GOOSE message exchange, the logical procedures that need to be followed are listed below:

- The source computes the hash for the PDU using a proper hash function;
- Using its own private key, the source digitally signs the hash previously generated adding it in the AutheticationValue GOOSE field;
- The receiver decrypt such an AuthenticationValue through source's public key;
- Through the same hash function, the receiver computes the hash for the received APDU;
- The receiver compares the two hash functions; if they match, the data integrity and the source identity is verified.

This mechanism allows for verification of the data integrity of the first eight octets by means of 16-bit-CRC, while the integrity of the remaining part is verified through the AuthenticationValue as well as the source identity.

4.2.1 Hash function, SHA and Digital Signature

A hash function is any function that can be used to map data of arbitrary size to data of fixed size. The values returned by a hash function are called hash values, hash codes, digests, or simply hashes. Hash functions accelerate table or database lookup by detecting duplicated records in a large file. An example is finding similar stretches in DNA sequences. They are also useful in cryptography. A cryptographic hash function allows one to easily verify that some input data maps to a given hash value, but if the input data is unknown, it is deliberately difficult to reconstruct it (or equivalent alternatives) by knowing the stored hash value. This is used for assuring integrity of transmitted data, and is the building block for HMACs, which provide message authentication.

The features of a generic hash function are listed below:

- The algorithm returns a fixed-length bit string composed by numbers and letters starting from an any dimension bit string;
- The algorithm is not invertible, thus the reconstruction of the original document starting from the digest is not possible. Therefore, each hash function is a unidirectional, not reversible function.

As the possible input combinations are way more numerous than the generable hashes, there must be at least one digest corresponding to one or more input string. Each time this occurs, the event is called collision.

The less collision can be found and the more a hash function is considered a good function.

Secure Hash Algorithm indicates a group of five different cryptographic hash functions developed starting from 1992 from the National Security Agency. As every hash function, the SHA produces a fixed-length digest starting from a variable-length input string. The algorithm belonging to such a group are called SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512, where the last four of them are also known as SHA-2. The first of them produces a digest equal to 160 bits, while the remaining ones return a digest whose length is defined by the SHA function's name (for example, SHA-512 produces a digest equal to 512 bits). The IEC 62351-6 requires the use of SHA-256 as hash function for the AuthenticationValue field.

Once the digest is generated through SHA-256, according to the IEC 62351-6, the hash value needs to be digitally signed by the sender of such a message by means of asymmetrical keys mechanism. A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or documents. A valid digital signature gives a recipient reason to believe that the message was created by a known sender (authentication), that the sender cannot deny having sent the message (non-repudiation), and that the message was not altered in transit (integrity).

A digital signature scheme typically consists of three algorithms:

- A key generation algorithm that selects a private key uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding public key.
- A signing algorithm that, given a message and a private key, produces a signature.
- A signature verifying algorithm that, given the message, public key and signature, either accepts or rejects the message's claim to authenticity.

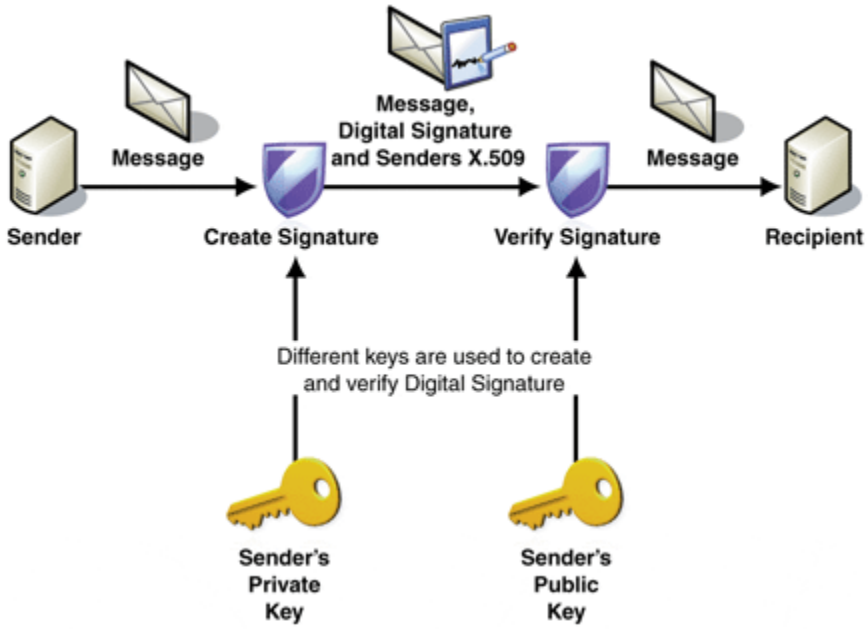


Figure 154 Digital Signature scheme

Two main properties are required. First, the authenticity of a signature generated from a fixed message and fixed private key can be verified by using the corresponding public key. Secondly, it should be computationally infeasible to generate a valid signature for a party without knowing that party's private key. A digital signature is an authentication mechanism that enables the creator of the message to attach a code that acts as a signature.

In IEC 61850, the sender signs the SHA-256 digest with a X.509 private key, while the receiver decrypts the message through the X.509 sender's public key. If no communication error has occurred, the receiver is able to decrypt the received message and verify the sender's identity. If the result of the process is different from the expectations, a communication error or a security issue must be assumed.

4.3 SECURITY TEST

The following attack was implemented as an ethical demonstration of security vulnerability inside the DITEN structure, at the University of Genoa; for this purpose, no VLAN networks were used and the configuration is shown in.

The attack consists of inserting a GOOSE message during a regular communication. Such a GOOSE contains intentionally altered parameters in such a manner that it is acknowledged as valid from the IED receiver.

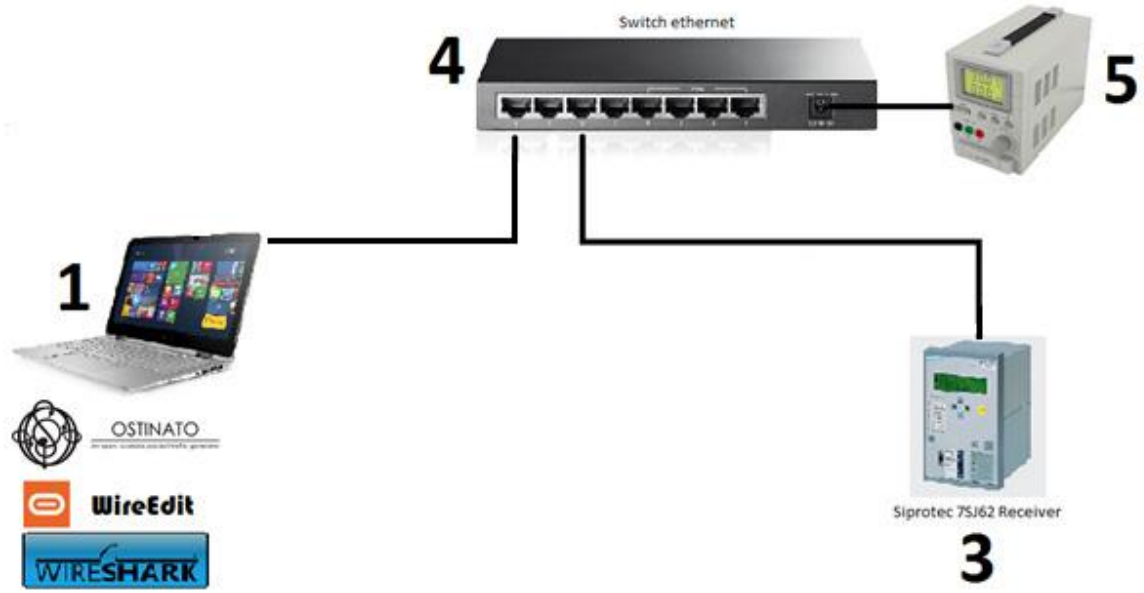


Figure 155 Security Test – Configuration

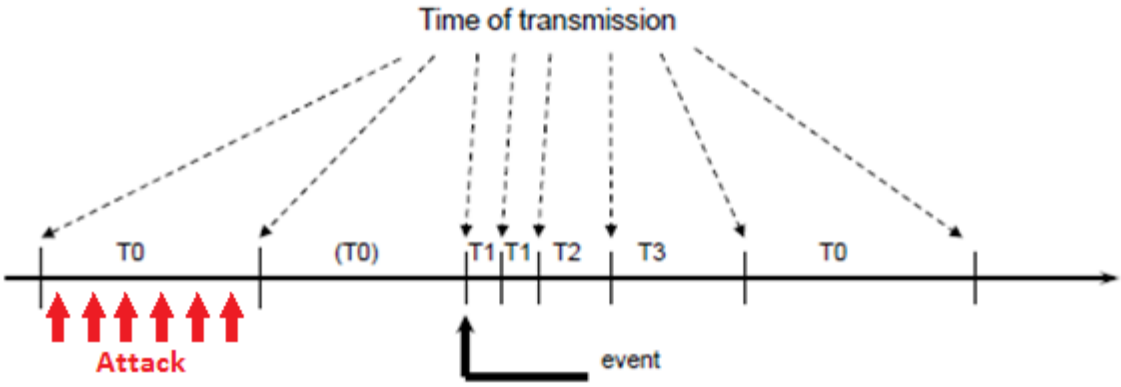


Figure 156 Security Test - Attack scheme

In order to send modified GOOSE, the PC (Device 1) was equipped with the needed software (Ostinato, WireEdit, and Wireshark) and the IED was configured as stated in 3.3.

In this case the retransmission scheme provides a telegram each 500ms and the altered GOOSE is sent between two regular messages (Figure 156), particularly the fourteenth and the fifteenth telegram (Figure 157).

The additional GOOSE (yellow square in Figure 157) was altered with the following parameters:

- Source MAC address: in order to indicate another device, it was modified to 04:50:5a:c0:40:08, but if MAC filtering function is carried out by the router, then the MAC needs to be put equal to one of the authorized devices by means of MAC spoofing
- State Number: this parameter should be equal to one of the expected, in this case the value 2 was used
- Sequence Number: it was put equal to 0
- Time Stamp: It was put one second ahead the other GOOSEs, particularly its value was modified to Dec 9, 2016 11:17:06 UTC
- Data: In order to perform an attack, this value should be switched to the opposite value from the previous one; in this case it was set to TRUE.

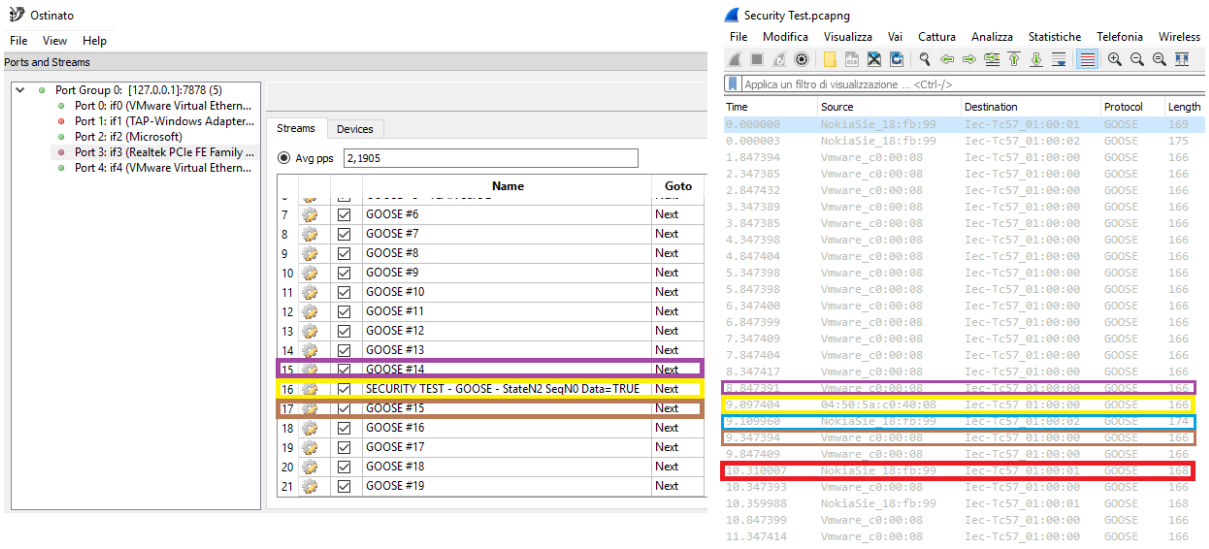


Figure 157 Security Test - Ostinato GUI on the left and Wireshark capture on the right

As Figure 157 shows, the SIPROTEC acknowledged the additional GOOSE, in fact a respond GOOSE was sent back. Starting from this point, the IED looked forward to

receive a valid GOOSE but, as the State Number was changed by the attacker, none of the received GOOSEs were valid anymore. After the TAL was elapsed, an error GOOSE was sent (red square in Figure 157).

A detailed view of such GOOSEs is provided by Figure 158.

Figure 158 Security Test - GOOSEs detailed view

The attacker does not directly know the high-level meaning of this GOOSE message (e.g. that this is a command from a circuit-breaker controller to a circuit breaker). However, they can decode the message to find and change the Data values.

Some classical IT techniques to prevent Ethernet layer 2 attacks could be applied to protect GOOSE messages. These practices include but are not limited to: set a dedicated VLAN ID for all trunk ports, disable unused ports and put them in an unused VLAN, use a dedicated VLAN for IEC 61850 devices, set all other ports to non-trunking, avoid the use of shared Ethernet such as WLANs or hubs.

At this level using the measures indicated the network is somewhat protected against intrusion originating from the outside.

In conclusion, it has been demonstrated that a simple attack enables an attacker to control IEC 61850-enabled control equipment. This control has the potential to cause outages that range from a single feeder on up. Meanwhile, it is of vital importance that the configuration of the network switch and routers be permitted solely for trusted traffic and users inside the substation network.

5. CONCLUSIONS

This thesis starts with an analysis of what is called Hybrid Ethernet-based Network. It consists of an Ethernet infrastructure used by different communication protocols together, each one with different functions, needs and application field. The implementation of a unique shared network allows to reduce cost of installation and maintenance using standard network components and standard cable infrastructure. A typical hybrid network is defined using two of the most popular communication protocols in their respective application field: Profinet for process automation and IEC 61850 for substation automation. After a deep analysis of these protocols, two network architectures are defined and tested. The results of the tests show that there is an approximated limit of usage of the bandwidth for additional traffic around 20%. Under this limit all the devices in the network run without losing their time performances and without operating failures. Moreover, the limit of 20% of bandwidth usage allows the network to run video surveillance tasks. A video streaming for surveillance can be overestimated with 1Mbps per camera: the available bandwidth allows to install a significant number of cameras on the network. After the results, Time Sensitive Networking is described. TSN is a state-of-the-art standard solution specifically designed to run both time-critical and non-time-critical protocols on the same network through the use of time synchronization, traffic scheduling and network configuration.

The implementation of hybrid networks potentially exposes the network itself to problems in terms of safety and security. The second part of this work aims to demonstrate that, from a technical point of view, IEC 61850 can be used for transferring safety related commands and signals. GOOSE messaging is tested with the communication errors specified in IEC 61874-3. Results shows that IEC 61850 supports the error detection and correction mechanisms that are typical of safety applications but today it is not appropriate to use IEC 61850 in safety application for the lack of certification of both: communication protocol and hardware. A commonly accepted protocol for testing the adequacy of 61850 based devices for safety applications will make possible the implementation of fully digital safe systems in the electrical word, with important benefits in terms of simplicity and dependability of the architectures.

The last part of this work is about security using IEC 61850. IEC 62351 specifies some security mechanisms that can be implemented to increase protocol security. After an analysis of the specifications for IEC 61850, a test on a non protected network is made in order to demonstrate how easy it is to perform an attack.

6. REFERENCES

[1] L. Rocca, P. Pinceti and M. Magro, "Can we use IEC 61850 for safety related functions?", Transactions on Environment and Electrical Engineering, vol. 1, no. 3, 2016. Available: 10.22149/teee.v1i3.22.

[2] M. Caserza Magro, P. Pinceti, L. Rocca and G. Rossi, "Safety related functions with IEC 61850 GOOSE messaging", International Journal of Electrical Power & Energy Systems, vol. 104, pp. 515-523, 2019. Available: 10.1016/j.ijepes.2018.07.033.

[3] Jasperneite, Juergen & Feld, J, “PROFINET: an integration platform for heterogeneous industrial communication systems.”, IEEE Conference on Emerging Technologies and Factory Automation. 1. 8 pp. - 822. 10.1109/ETFA.2005.1612610.

[4] P.A. Laplante, “Real-Time systems design and analysis”, Wiley-IEEE Press, 2004.

[5] T. Frank, K. Eckert, T. Hadlich, A. Fay, C. Diedrich, B. Vogel-Heuser – “Workflow and decision support for the design of distributed automation systems”, 10th IEEE International Conference on Industrial Informatics (INDIN), 2012, ISBN: 978-1-4673-0312-5.

[6] P. Ferrari, A. Flammini, D. Marioli, A. Taroni, “Experimental evaluation of PROFINET performance”, IEEE Workshop on Factory Automation Systems, 2004, pp.331-334.

[7] M.Caserza Magro, P. Pinceti, “Measuring real time performances of PC-based industrial control systems”, IEEE Conference on Emerging Technologies and Factory Automation, 2007, pp.540-547.

[8] P. Ferrari, A. Flammini, S. Rinaldi, G. Prytz, “Mixing real time Ethernet traffic on the IEC 61850 process bus”, Int. Workshop on Factory Communication Systems, 2012, pp. 153-156.

[9] L. Seno, P. Vitturi, “Real Time Ethernet Networks Evaluation Using Performance Indicators”, IEEE Conference on Emerging Technologies and Factory Automation, 2009, pp. 1-8.

[10] G. Creech, "Black Channel Communication: What is it and How Does it Work?", Measurement and Control, vol. 40, no. 10, pp. 304-309, 2007. Available: 10.1177/002029400704001003.

[11] C. Kriger, S. Behardien and J. Retonda-Modiya, "A Detailed Analysis of the Generic Object-Oriented Substation Event Message Structure in an IEC 61850 Standard-Based Substation Automation System", International Journal of Computers Communications & Control, vol. 8, no. 5, p. 708, 2013. Available: 10.15837/ijccc.2013.5.329.

[12] K.P. Brand and M. Ostertag, "Safety related, distributed functions in substations and the standard IEC 61850", Bologna Power Tech Conference, 2003.

[13] Members of The MTL Instruments Group, “An introduction to Functional Safety and IEC 61508”, AN9025, 2002.

[14] Ladkin, Peter B. and Causalis. “An Overview of IEC 61508 on E / E / PE Functional Safety.”, 2008.

[15] R. Hunt and B. Popescu, “Comparison of PRP and HSR Networks for Protection and Control Applications”, Western Protective Relay Conference 2015.

[16] Hoyos, Juan et al. “Exploiting the GOOSE protocol: A practical attack on cyber-infrastructure.” 2012 IEEE Globecom Workshops (2012): 1508-1513.

[17] "Functions of LLC and MAC sub-layers of Data Link Layer | Computer Networking Demystified", Computernetworkingsimplified.in, 2019. [Online]. Available: http://computernetworkingsimplified.in/data-link-layer/components-data-link-layer-llc-mac/.

[18] IEC 61784-3, “Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions”, 2016

[19] ABB, “IEC 61850 Edition 2 Communication Protocol Manual”, 2014.

[20] ABB, “IEC 61850 Engineering Guide: 615 series ANSI”, 2011.

[21] IEC 61850-1, “Communication networks and systems for power utility automation - Part 1: Introduction and overview”, 2013.

[22] IEC 61850-7-1, “Communication networks and systems for power utility automation - Part 7-1: Basic communication structure - Principles and models”, 2011.

[23] IEC 61850-7-2, “Communication networks and systems for power utility automation - Part 7-2: Basic information and communication structure - Abstract communication service interface (ACSI)”, 2010.

[24] IEC 61850-8-1, “Communication networks and systems for power utility automation - Part 8-1: Specific communication service mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3”, 2011.

[25] M. Adamiak, D. Baigent and R. Mackiewicz. “IEC 61850 Communication Networks and Systems in Substations : An Overview for Users.”, The Protection & Control Journal, 61-68, 2009.

[26] IEC 62351-6, “Power systems management and associated information exchange - Data and communications security - Part 6: Security for IEC 61850”, 2007.

[27] Jin Cheol Kim and Tae Hun Kim, “Implementation of secure IEC61850 communication”, CIRED Workshop, June 2014.

[28] A. Ingrey, P. Lerévérend and A. Hildebrandt, “ Manual Safety Integrity Level”, Pepperl+Fuchs, 2017.

[29] Talha Abdul Rashid, Muhammad & Yussof, Salman & Yusoff, Yunus, “Trust System Architecture for Securing GOOSE Communication in IEC 61850 Substation Network”, International Journal of Security and Its Applications. 10. 289-302. 10.14257/ijsia.2016.10.4.27.